Murdock 1976

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REAL TIME SIMULATION AND CONTROL
3000 TON SURFACE EFFECT SHIP
WITH NEGATIVE DRAG CHARACTERISTICS
IN SEA STATE

by

Lee Lewis Oliphant

December 1980

Thesis Advisor: A. Gerba

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Real Time Simulation and Control 3000 Ton Surface Effect Ship With Negative Drag Characteristics In Sea State		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lee Lewis Oliphant		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 117
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Surface Effect Ship; Real Time Simulation; Negative Drag Characteristic; Speed Control; Sea State Effect

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The model of a Surface Effect Ship was refined to include simplified propulsion dynamics, negative drag characteristics, sea state effects and an autopilot for speed control. These design modifications were introduced into a real time, man controlled simulation of a 3000 ton Surface Effect Ship (3K-SES) in 5 degrees of freedom (RTS5D) and results were compared with a Data Base Program (DBSIM5D) based on towing tank data scaled up to model the ship.

Hardware and Software design changes were incorporated into the RTS5D model to provide a more accurate approximation of real time, a faster computer iteration time, and a broach condition warning if the operator exceeded certain thrust vectoring limits.

Real Time Simulation and Control 3000 Ton Surface Effect Ship With Negative Drag Characteristics In Sea State

by

Lee Lewis Oliphant
Lieutenant, United States Navy
B.S., University of Texas, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1980

ABSTRACT

The model of a Surface Effect Ship was refined to include simplified propulsion dynamics, negative drag characteristics, sea state effects and an autopilot for speed control. These design modifications were introduced into a real time, man controlled simulation of a 3000 ton Surface Effect Ship (3K-SES) in 5 degrees of freedom (RTS5D) and results were compared with a Data Base Program (DBSIM5D) based on towing tank data scaled up to model the ship.

Hardware and Software design changes were incorporated into the RTS5D model to provide a more accurate approximation of real time, a faster computer iteration time, and a broach condition warning if the operator exceeded certain thrust vectoring limits.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A_{ws}	Average sidewall wetted area, starboard side	ft^2
A_{wp}	Average sidewall wetted area, port side	ft^2
A_{31}	Added mass coefficient in roll force equation	ft-slug
A_{33}	Added mass coefficient in pitch force equation	ft-slug
A_{w2}	Average wetted sidewall area of the bow	ft^2
A_{w1}	Average wetted sidewall area of the stern	ft^2
A_{22}	Added mass coefficient in yaw force equation	slug $\frac{s^2}{ft}$ $lb_f/$
$\dot{\beta}$	Sideslip angle	rad
$C_{DX_{(K)}}$	Coefficients drag in x-direction	$lb_f s^2 lb_m/ft^2$
C_{DY}	Coefficient drag in y-direction	$lb_f s^2 lb_m/ft^2$
C_{DZP}	Sidewall roll moment lumped parameter coefficient	$lb_f s^2/ft^2$
C_{DP}	Bow pitch force lumped parameter coefficient	non-dimensional
F_{sw}	Sidewall roll force	lbs
F_{ss}	Sidewall starboard buoyancy force	lbs
F_{sp}	Sidewall port buoyancy force	lbs
F_1	Stern buoyancy force	lbs
F_2	Bow seal pitching force	lbs
F_3	Bow buoyancy force	lbs
g	Gravitational acceleration	ft/s^2

I_X	Moment of inertia about x-axis	slug ft^2
I_Y	Moment of intertia about y-axis	slug ft^2
I_Z	Moment of inertia about z-axis	slug ft^2
K	Summation of moments about x-axis	lbs-ft
ℓ_{sw}	Length of sidewall	ft
ℓ_{dp}	Actual draft of port sidewall	ft
ℓ_w	x-direction displacement of hull drag centroid	ft
ℓ_{d_1}	Average draft of SES sidewall	ft
ℓ_d	Average draft of bow seal	ft
ℓ_x	Length from center of gravity to stern	
ℓ_{x_1}	Average draft of widewall	ft
ℓ_{31}	Pitch moment lever arm for bow sidewall buoyance force	ft
ℓ_3	Pitch moment lever arm for bow seal force	ft
m	Mass of the rigid ship	ft
M	Summation of moments about y-axis	lbs-ft
N	Summation of moments about z-axis	lbs-ft
P_b	Plenum pressure	lbs-ft^2
p	Lumped drag centroid point	non-dimensional
p'	Lumped drag centroid point	non-dimensional
$\dot{p}$	Roll acceleration	rad/sec^2

q	Pitch rate	rad/sec
$\dot{q}$	Pitch acceleration	rad/sec^2
r	Yaw rate	rad/sec
$\dot{r}$	Yaw acceleration	rad/sec^2
s_1	Turning moment lever arm of no. 1 engine	ft
s_2	Turning moment lever arm of no. 2 engine	ft
s_3	Turning moment lever arm of no. 3 engine	ft
s_4	Turning moment lever arm of no. 4 engine	ft
T_7	Total thrust magnitude on no. 1 engine	lbs
T_8	Total thrust magnitude on no. 2 engine	lbs
T_9	Total thrust magnitude on no. 3 engine	lbs
T_{10}	Total thrust magnitude on no. 4 engine	lbs
T_{forw}	Total forward thrust vector of effectors	lbs
T_{side}	Total side thrust vector of effectors	lbs
T_{yaw}	Total turning moment generated by effectors	lbs ft
u	Velocity in x-direction (surge)	ft/sec
$\dot{u}$	Acceleration in x-direction	ft/sec^2
v	Velocity in y-direction (sway)	ft/sec
$\dot{v}$	Acceleration in y-direction	ft/sec^2
v_s	Total velocity	ft/sec

w_e	Width of bow seal	ft
X	Summation of forces in x-direction	lbs
X_o	X_{NAV} coordinate of SES	ft
$\dot{X}_o$	X_{NAV} velocity of SES	ft/sec
Y	Summation of forces in y-direction	lbs
Y_o	Y_{NAV} coordinates of SES	ft
$\dot{Y}_o$	Y_{NAV} velocity of SES	ft/sec
δ	Effector angle commanded	rad
δ_7	Effector angle of no. 1 nozzle	rad
δ_8	Effector angle of no. 2 nozzle	rad
δ_9	Effector angle of no. 3 nozzle	rad
δ_{10}	Effector angle of no. 4 nozzle	rad
ψ	Heading angle of SES	rad
ϕ	Roll angle of SES	rad
θ	Pitch angle of SES	rad
ρ	Density of sea water	lbm/ft^3

I. INTRODUCTION

Rising fuel costs, critical manpower shortages and an increasing need for rapid deployment of forces in defense of United States' interests throughout the world mandate the development of fuel efficient, high speed, minimum manned ships for the Navy. The Captured Air Buddle (CAB) Surface Effect Ship (SES) holds great promise in providing an answer to these problems.

A ship with minimum manning traveling at the high speeds envisioned for the SES would require highly skilled operator personnel. It would be necessary to provide these personnel with real time operational training similar to that which pilots receive in aircraft simulators. This would test the man in responding to failure situations that may be encountered during actual operation and train personnel in control techniques that are unique to SES type vehicles. A Real Time Simulator could also be used as a means to test hardware used in SES craft and provide a design tool for the development of future modifications.

A simplified, non-linear five degree of freedom Real Time Simulator (RTS5D) for the 3K-SES has been developed by T. S. Nelson [Ref. 1] such that a continuously observable real time solution is interfaced with man-generated control. Certain "man-in-the-loop" experiments have been conducted using the

RTS5D that have demonstrated limitations in that model. It was necessary to: 1) refine ship dynamics; 2) introduce simplified propulsion dynamics; 3) constrain thrust angle in order to remain within the bounds of programmed dynamic response (avoid broaching); 4) refine the software to provide faster computer iteration time; and 5) change the hardware to provide the operator with speed control as well as turning control.

Accordingly, changes have been made in the RTS5D model to more accurately reflect the nonlinear forward drag characteristics of the SES. A speed controller has been designed to allow operation of the 3K-SES within a speed range of 40-60 knots and limits have been placed on the thruster deflection angle in order to prevent broaching, a condition which is not allowed under the existing dynamic model. Software refinements have been incorporated to improve computer iteration time.

II. EQUATIONS OF MOTION

A. COORDINATE SYSTEMS AND ASSUMPTIONS

The RTS5D model of the 3K-SES used simplified equations of motion. For completeness of comparison the development of these equations is repeated here. The rational for requiring simplification was to start with differential equations of motion that could be utilized effectively in a real time simulation. To satisfy this requirement a model was selected that would use point source of drag forces consisting of only a few lumped-coefficient terms. The equations used are based upon those developed in Ref. 2 which follow this concept. Additional assumptions used in the simplified 5 DOF equations were as follows:

1) All accelerations are measured at the center of gravity.
2) Vehicle is "free-to-heave".
3) All cross coupled moments of inertia are zero.
4) All cross products of angular velocities in force equations are zero.
5) All roll and pitch angles used in force equations are subject to small angle approximations.
6) Calm water conditions.
7) Mass and mass distribution of vehicle are constant.
8) Effect of changes of aerodynamic force are neglected.

Utilizing the above assumptions, the simplified 5 DOF equations of motion are developed using the coordinate systems shown in Fig. 1 and Fig. 2, the force vector diagrams shown in Fig. 3, Fig. 4A, Fig. 4B, and the following equations from Newton's laws of motion:

SURGE	$m(\dot{u} - vr)$	$= X$
SWAY	$m(\dot{v} - ur)$	$= Y$
YAW	$I_Z \dot{r}$	$= N$
PITCH	$I_Y \dot{q}$	$= M$
ROLL	$I_X \dot{p}$	$= K$

where:

m	= mass of the rigid ship	' slugs
u	= velocity in x-direction (SURGE)	' ft/sec
v	= velocity in y-direction (SWAY)	' ft/sec
r	= angular velocity about the z-axis	' deg/sec
p	= angular velocity about the x-axis	' deg/sec
q	= angular velocity about the y-axis	' deg/sec
I_Z	= moment of inertia about the z-axis	' $lb_m\text{-}ft^2$
I_Y	= moment of inertia about the y-axis	' $lb_m\text{-}ft^2$
I_X	= moment of inertia about the x-axis	' $lb_m\text{-}ft^2$
X	= summation of forces in x-direction	' lb_f
Y	= summation of forces in y-direction	' lb_f
N	= summation of moments about the z-axis	' $ft\text{-}lb_f$
M	= summation of moments about the y-axis	' $ft\text{-}lb_f$
K	= summation of moments about the x-axis	' $ft\text{-}lb_f$

Additionally the following navigation relationships are utilized (see Fig. 1).

$$\dot{X}_o = u \cos\psi - v \sin\psi$$

$$\dot{Y}_o = u \sin\psi + v \cos\psi$$

$$V_s^2 = u^2 + v^2$$

$$\beta = \tan^{-1} (-v/u)$$

where:

ψ = heading angle

β = drift angle

δ = thrust vector angle

Also,

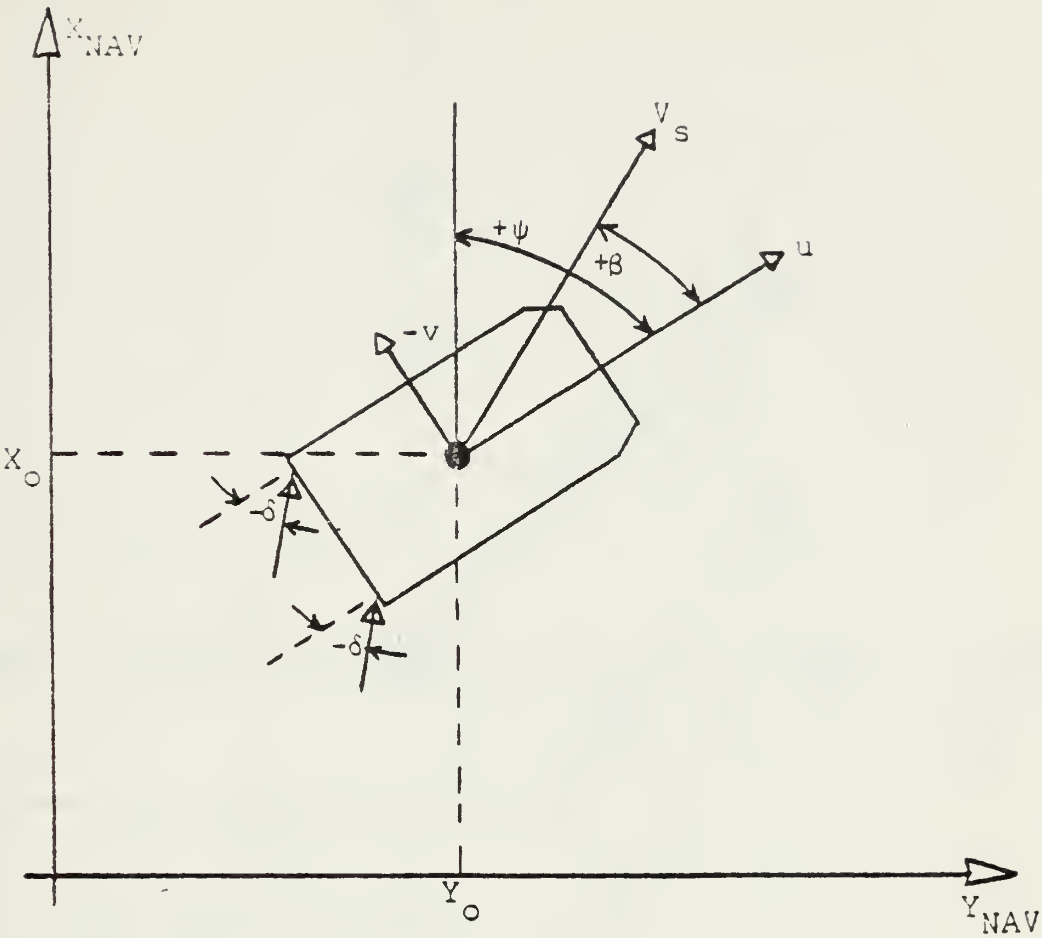
$$X_o = \int \dot{X}_o \, dt \qquad u = \int \dot{u} \, dt$$

$$Y_o = \int \dot{Y}_o \, dt \qquad v = \int \dot{v} \, dt$$

$$r = \int \dot{r} \, dt \qquad \psi = \int r \, dt$$

$$p = \int \dot{p} \, dt \qquad \phi = \int p \, dt$$

$$q = \int \dot{q} \, dt \qquad \theta = \int q \, dt$$

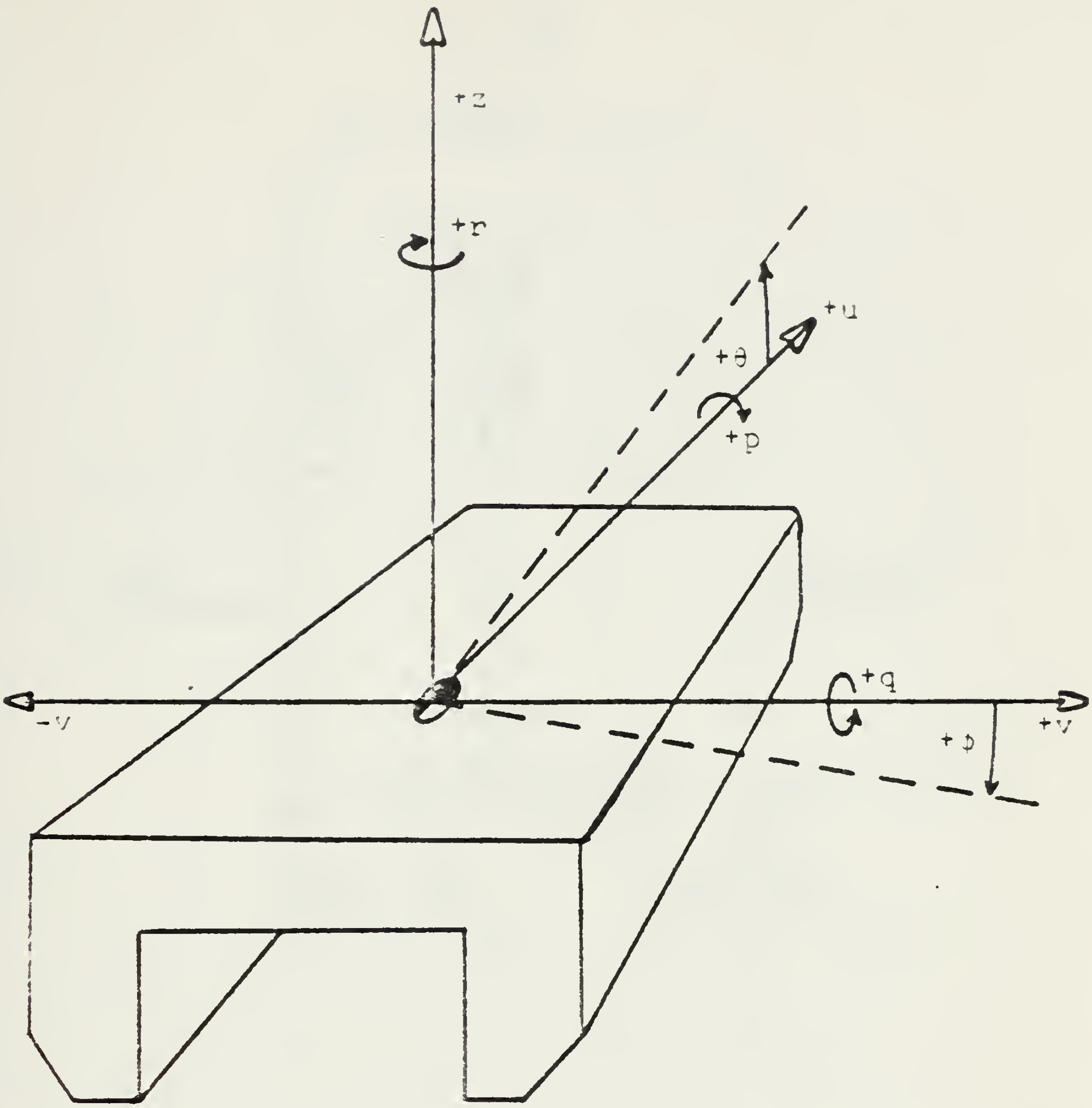


Note: 1) ψ=0 when vehicle heads paralled to + X_{NAV} axis.

2) Vehicle in figure above shown with negative thrust vector angle δ, positive ψ, negative β, negative sway velocity v, i.e., in a right turn

Figure 1

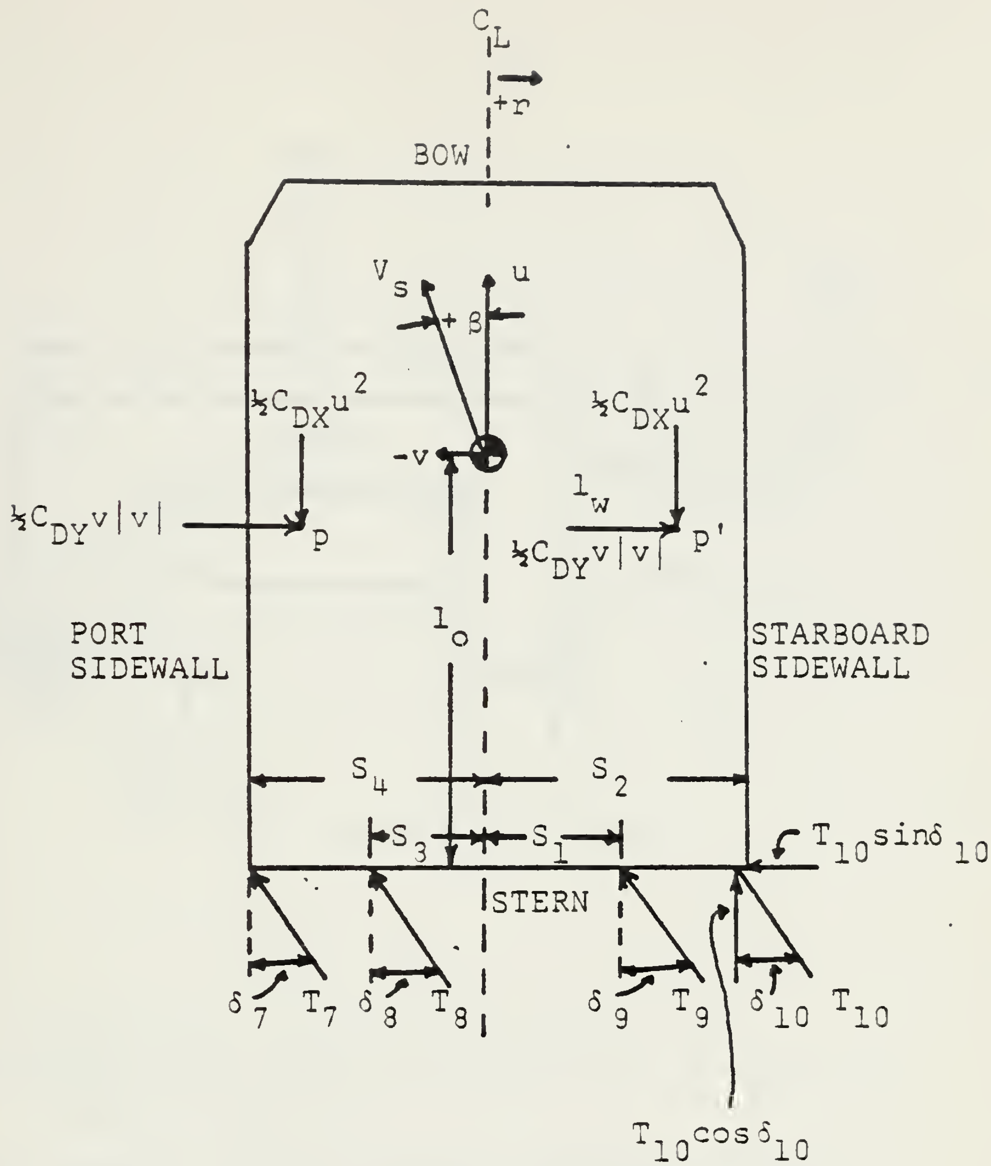
Definition Of Coordinate System (Part I)



Note: Direction for positive velocity and angular rates are shown

Figure 2

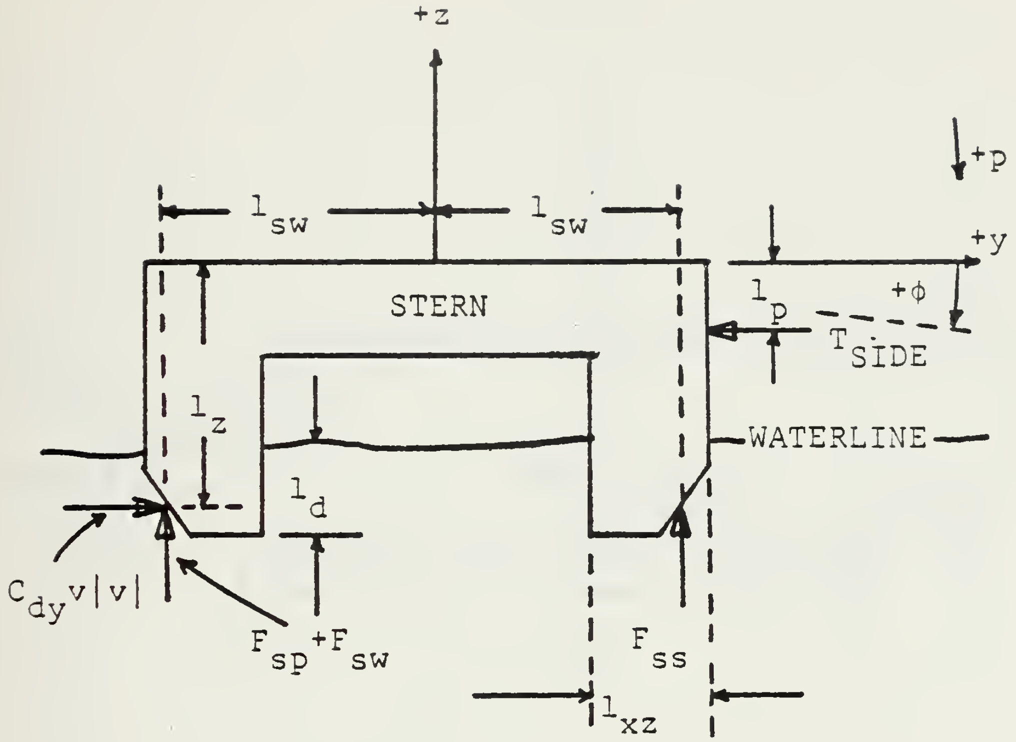
Definition Of Coordinate System (Part II)



Note:
1) p and p' are equivalent point force centroids.
2) δ_7, δ_8, δ_9, δ_{10} are thrust vector angles and are not required to be equal.
3) T_7, T_8, T_9, T_{10} are thrust magnitudes and are not required to be equal.
4) Force directions shown are for a right turn

Figure 3

Surface Effect Ship (Top View)

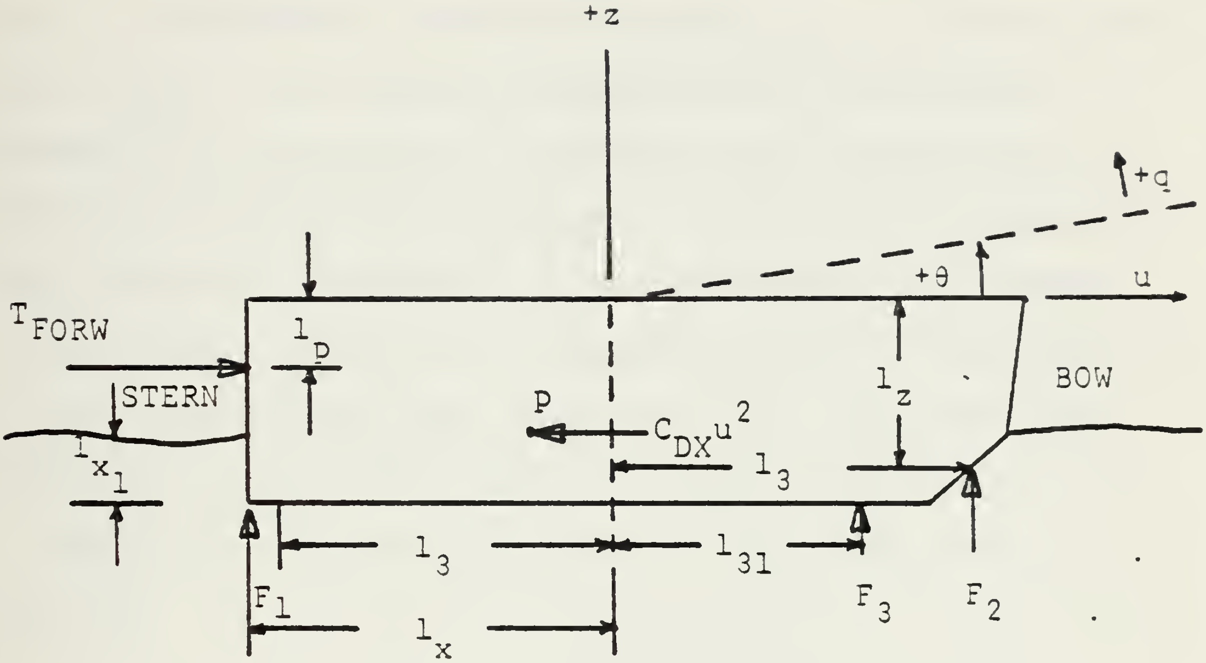


Note: 1) Right Turn Forces Acting, i.e., δ_7, δ_8, δ_9, δ_{10} are negative

2) $T_{SIDE} = T_7 \sin\delta_y + T_8 \sin\delta_8 + T_9 \sin\delta_9 + T_{10} \sin\delta_{10}$

Figure 4A

Surface Effect Ship (Stern View)



Note: $T_{FORW} = T_7 \cos\delta_7 + T_8 \cos\delta_8 + T_9 \cos\delta_9 + T_{10} \cos\delta_{10}$
(see Fig. 3)

Figure 4B

Surface Effect Ship (Side View)

B. FORCES AND MOMENTS

1. Surge Forces (see Fig. 4B)

The forward acceleration was determined to be generated by the summation of thrust vectored in the forward direction T_{forw}. The resultant acceleration of the SES was assumed to be opposed by a retarding force exhibiting a "velocity-squared" characteristic. Additionally, Newton's laws of motion specified a "v.r" product that contributed to the drag component of the equation when the vehicle was in a turn (contrifugal force reaction term). The simplified equation of motion for the surge acceleration follows from a summation of forces in the u direction as depicted in Fig. 4B.

$$\dot{u} = (T_{forw}/m) - (C_{DX}u^2/m) + vr$$

2. Sway Forces (see Fig. 4A)

The side acceleration of the vehicle was analyzed to be the result of the summation of the thrust vectored parallel to the vehicle's y-axis and a retardation "velocity squared" phenomena such as that produced by the cross flow drag term of the sidewall as described in Ref. 3. This retardation force is augmented by a "u·r" product as specified by Newton's Law.

$$\dot{v} = T_{side}/m - C_{DY}\, v|v|/m - ur$$

Note the requirement to force the $C_{DY}v^2/m$ term to maintain the sign of the sway velocity such that both left and right turns may be computed.

3. Yaw Moments (see Fig. 3)

The yaw acceleration was a turning moment summation which was defined to be influenced by three primary forces operating over moment arms. It is of interest to note that the third term is an "added mass" term which is considered to operate on the same pressure point as the $C_{DY}v|v|$ term. The term T_{yaw} is a summation of moments generated by the four thrusters as shown in Fig. 3.

$$T_{yaw} = s_4T_7 \cos\delta_7 + s_3T_8 \cos \delta_8 - s_1T_9 \cos \delta_9$$
$$- s_2T_{10} \cos \delta_{10} - \ell_oT_7 \sin \delta_7 - \ell_oT_8 \sin \delta_8$$
$$- \ell_oT_9 \sin \delta_9 - \ell_oT_{10} \sin \delta_{10}$$

$$\dot{r} = C_{DY}\, \ell_w\, v|v|/I_Z + A_{22}uv\ell_w/I_Z + T_{YAW}/I_Z$$

4. Pitch Moments (see Fig. 4B)

The pitch acceleration was assumed to be the summation of moments generated by forward thrust, T_{forw}, the buoyancy of the sidewalls of the SES, F_1 and F_2 and a vertical force generated at the bow of the vehicle, F_3. This vertical bow force was defined to be a lumped parameter term which modeled the reaction force due to plenum pressure acting against the bow seal.

ℓ_d = average draft of bow seal

A_{w_1} = $\ell_x \ell_{x_1} + ((\ell_x \tan\theta)/2)\ell_x$

A_{w_2} = $\ell_x \ell_{x_1} - ((\ell_x \tan\theta)/2\ell_x$

F_1 = $A_{w_1} \ell_{x_2} \rho g$

F_2 = $A_{w_2} \ell_{x_2} \rho g$

F_3 = $C_{DP}\bar{p}_b w_e(\ell_d - \ell_{31} \tan\theta)$

ℓ_{x1} = average draft of sidewall

p_b = plenum pressure

A_{w_1} = average wetted sidewall area of the stern

ℓ_{x2} = width of one sidewall

w_e = width of bow seal

A_{w_2} = average wetted sidewall area of the bow

A_{33} = added mass coefficient

ρ = water density

g = 32.3 ft/s^2

ℓ_{31} = lever arm of bow seal

ℓ_3 = lever arm of buoyancy force

$$\dot{q} = (T_{forw}\, \ell_p + F_3\ell_{31} + F_2\ell_3 - C_{DX}\, u^2\, \ell_z$$
$$- F_1\ell_3 - A_{33}uq)/I_Y$$

Note the added mass term $A_{33}uq$ which was required to provide damping to the pitch moment and is specified in Ref. 2.

It was found that the response of the SES to a pitch perturbation without the added mass component was undamped and approximately sinusoidal. Unlike the yaw equation where the added mass term was a small contributor to damping, the pitch added mass term was found to be of significant importance in modeling the known pitch motion.

5. Roll Moments (see Fig. 4A)

The roll acceleration equation was analyzed to be a summation of moments generated by buoyancy forces of the port and starboard sidewalls, F_{sp} and F_{ss}, the thrust vectored parallel to the y-axis of the vehicle, T_{side}, a side force $C_{DY}\, v|v|$, and a lumped coefficient vertical force, F_{sw}. The vertical force F_{sw} was defined as the force generated in a turn due to the sidewall curvature (dead rise angle) acting against the cross flow of water.

$$F_{sp} = \rho g\, A_{wp}\, \ell dp$$

$$F_{ss} = \rho g\, A_{ws}\, \ell ds$$

$$F_{sw} = C_{DZP}\, v|v|$$

ℓ_{d_1} = average draft of SES sidewall

$$\ell_{dp} = \ell_{d_1} - \ell_{sw} \tan \phi$$

$$\ell_{ds} = \ell_{d_1} + \ell_{sw} \tan \phi$$

A_{wp} = average wetted area port

A_{ws} = average wetted area starboard

$$\dot{p} = ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\ell_p + C_{DY}\, v|v|\, \ell_z$$

$$- F_{sw}\ell_{sw} - A_{31}\, up)/I_X$$

Again note the added mass term $A_{33}up/I_X$ which was found to be essential in modeling the damping phenomena of the roll motion.

In summary, the simplified 5 DOF equations of motion used in RTS5D for the 3K TON SES are:

<u>SURGE</u> $\dot{u} = T_{forw}/m - C_{DX}\, u^2/m + vr$ (Fig. 4B)

<u>SWAY</u> $\dot{v} = T_{side}/m - C_{DY}\, v|v|/m = ur$ (Fig. 4A)

<u>YAW</u> $\dot{r} = T_{yaw}/I_Z + C_{DY}\, v|v|\, \ell_w/I_Z + A_{22}uv\ell_w/I_Z$

(Fig. 3)

<u>PITCH</u> $\dot{q} = (T_{forw}\ell_p + F_3\ell_{31} + F_2\ell_3 - C_{DX}\, u^2\ell z$

$- F_1\ell_3 - A_{34}uq)/I_Y$ (Fig. 4B)

<u>ROLL</u> $\dot{p} = ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\ell_p + C_{DY}\, v|v|\ell_z$

$- F_{sw}\ell_{sw} - A_{33}\, up)/I_X$ (Fig. 4A)

C. PARAMETER IDENTIFICATION

These equations are an extension of the 3 DOF flat turn SES model developed by Gerba and Thaler in Ref. 4. The identification of craft parameters C_{DX}, C_{DY}, and ℓ_w is described in Ref. 4 and repeated here for completeness.

The surge drag coefficient C_{DX} is determined by selecting a steady state turn condition, where T_{forw}, u, v, r, and m are known.

$$\dot{u} = 0 = \frac{T_{forw}}{m} - \frac{C_{DX}u^2}{m} + vr$$

from which

$$C_{DX} = \frac{T_{forw} + mvr}{u^2}$$

The sway drag coefficient C_{DY} is determined by the same steady state turn condition which requires

$$\dot{v} = 0 = \frac{T_{side}}{m} - \frac{C_{DY}}{m} v|v| - ur$$

where

$$C_{DY} = \frac{T_{side} - mur}{v|v|}$$

The sway drag moment arm follows utilizing the yaw acceleration equation

$$\dot{r} = 0 = \frac{T_{yaw}}{I_Z} + \frac{C_{DY}v|v|\ell_w}{I_Z} + \frac{A_{22}\ uv\ell_w}{I_Z}$$

which yields

$$w = \frac{-T_{yaw}}{C_{DY}\ v|v| + A_{22}uv}$$

The addition of the pitch and roll equations introduces additional lumped coefficients C_{DP} and C_{DZP}. These are easily solved using known constants ℓ_w, ℓ_p, C_{DY}, ℓ_z, ℓ_{sw}, w_e, p_b, ℓ_d and steady state values of F_{sp}, F_{ss}, F_1, F_2, T_{forw}, T_{side}, u, v, and r. It is significant to note that in a steady state condition the angular roll rate, p, and angular pitch rate, q, are both required to equal zero. This is in contrast to the yaw rate, r, where a finite steady state value is desired in a turn. Thus the added mass terms of $A_{33}uq$ and A_{31} up in the pitch and roll equations are not utilized in the determination of craft parameters; their function is strictly confined to damping of their respective accelerations. Therefore, utilizing the pitch equation with known steady state turn values.

$$\dot{q} = 0 = (T_{forw}\ell_p + C_{DP}p_b w_e(\ell_d - \ell_{31} \tan \theta)\ell_{31}$$
$$+ F_2\ell_3 - C_{DX}u^2\ell 2 - F_1\ell_3)/I_Y$$

yields

$$C_{DP} = \frac{C_{DX}u^2\ell_z + F_1\ell_3 - F_2\ell_3 - T_{forw}\ \ell_p}{p_b w_e(\ell_d - \ell_{31} \tan \theta)\ell_{31}}$$

The roll equation under steady state turn conditions yields

$$\dot{p} = 0 = ((F_{sp} - F_{ss})\ell_w - T_{side}\ \ell_p + C_{DY}\ v|v|\ \ell z$$
$$- C_{DZP}\ v|v|\ \ell_{sw})I_Z$$

from which

$$C_{DZP} = ((F_{sp} - F_{ss})\ell_w\ T_{side}\ell_p + C_{DY}\ v_{ss}|v_{ss}|\ell_z)/$$
$$(v_{ss}|v_{ss}|\ell_{sw})$$

where

F_{sp}	= port sidewall buoyancy in pounds	(Fig. 4A)
F_{ss}	= starboard sidewall buoyancy in pounds	(Fig. 4A)
T_{side}	= thrust vectored parallel to vehicles y-axis in pounds	(Fig. 3)
v_{ss}	= sway steady state velocity in feet/sec	
C_{DY}	= sway drag coefficient for steady state condition	
ℓ_w	= sway drag moment in feet	(Fig. 3)
ℓ_p	= effector thrust moment arm in feet	(Fig. 4A)
ℓ_z	= surge drag moment arm in feet	(Fig. 4A)
ℓ_{sw}	= sidewall buoyancy force moment arm in feet	(Fig. 4A)

III. RTS5D VALIDATION

A. DBSIM5D BENCHMARK

Validation of the RTS5D was attempted using the data base program (DBSIM5D) [Ref. 5] as a benchmark. However, new guidance in the proper use of the data base program contained in Ref. 6 revealed the following errors in the use of the DBSIM5D. 1) Initial conditions had been improperly set and 2) Data used for validation for the 56 knot run at effector deflection angles greater than 10° was not correct because of the thrust inlet broach conditions which existed using the DBSIM5D. These conditions were not accounted for in the RTS5D model. The noted discrepancies in the operation of the DBSIM5D necessitated a revalidation of the RTS5D.

B. VALIDATION RESULTS

Using the information contained in Ref. 6, the DBSIM5D program was again used as a benchmark to validate the RTS5D simulation. A DBSIM5D sequence of runs with initial forward velocities equal to 40 knots, 50 knots and 60 knots with step effector angles of 5° and 10° (at 40 knots), 5°, 10° and 15° (at 50 knots), and 5° (at 60 knots) were compared to identical runs of the RTS5D. Thrust effector angles were kept below a maximum that would have caused a broach

condition. Broach conditions are explained in Chapter IV. The first peak overshoot value and "Quasi Steady State" values for the variables u (surge), v (sway), r (yaw rate), ϕ (roll), and θ (pitch) were used as a basis for comparison. The results are shown in Tables I-III. Significant error in forward velocity occurred in the 50 knot test. Also, it was noted that in 40 knot test with 5° thruster deflection angle that the DBSIM5D actually reached a "Quasi Steady State" forward velocity greater than the initial value upon entering the turn. These conditions led to the development of the model of the RTS5D with negative drag characteristics in sea state. Differences in the other measured parameters were addressed by parameter adjustment on the new model in order to more closely match values obtained with the DBSIM5D.

TABLE I

RTS5D and DBSIM5D Performance Test at 40 Knots

effector angle	DBSIM5D 1st pk	DBSIM5D Qss	RTS5D 1st pk	RTS5D Qss
	ϕ in degrees		ϕ	
5°	1.01	.742	.17	.12
10°	1.46	1.39	.31	.24
	ϕ in degrees		ϕ	
	1st pk	Qss	1st pk	Qss
5°	1.44	1.40	1.27	1.21
10°	1.48	1.48	1.28	1.20
	u in ft/sec		u	
	1st pk	Qss	1st pk	Qss
5°	67.4	70.9	n/a	67.03
10°	n/a	66.3	n/a	65.98
	v in ft/sec		v	
	1st pk	Qss	1st pk	Qss
5°	1.42	1.37	2.84	2.42
10°	3.20	3.26	3.83	3.41
	r in degrees/sec		r	
	1st pk	Qss	1st pk	Qss
5°	.396	.403	.58	.33
10°	.953	.959	.98	.66

TABLE II

RTS5D and DBSIM5D Performance Test at 50 Knots

effector angle	DBSIM5D 1st pk	DBSIM5D Qss	RTS5D 1st pk	RTS5D Qss
	ϕ in degrees		ϕ	
5°	.424	.446	.27	.19
10°	.757	.886	.49	.38
15°	1.02	1.4	.72	.57
	θ in degrees		θ	
5°	.862	.892	1.16	1.15
10°	.943	1.13	1.15	1.16
15°	1.12	1.35	1.22	1.17
	u in ft/sec		u	
5°	n/a	83.6	n/a	83.8
10°	n/a	78.7	n/a	82.51
15°	n/a	69.2	n/a	80.58
	v in ft/sec		v	
5°	1.31	1.30	3.55	3.02
10°	2.72	2.70	4.79	4.26
15°	4.05	4.33	5.69	5.207
	r in degrees/sec		r	
5°	.368	.365	.73	.41
10°	.864	.870	1.22	.82
15°	1.35	1.44	1.66	1.26

TABLE III

RTS5D and DBSIM5D Performance Test at 60 Knots

effector angle	DBSIM5D		RTS5D	
	ϕ	in degrees	ϕ	
	1st pk	Qss	1st pk	Qss
3°	.141	.200	.39	.274
	θ	in degrees	θ	
	1st pk	Qss	1st pk	Qss
5°	.521	.583	1.14	1.07
	u	in ft/sec	u	
	1st pk	Qss	1st pk	Qss
5°	n/a	97.8	n/a	100.29
	v	in ft/sec	v	
	1st pk	Qss	1st pk	Qss
5°	2.03	1.84	4.26	3.62
	r	in degrees/sec	r	
	1st pk	Qss	1st pk	Qss
5°	.532	.494	.87	.487

IV. MODEL DEVELOPMENT

A. DRAG CHARACTERISTICS

1. Surge Equation Model

The RTS5D model assumed a velocity squared drag retarding characteristic for the 3K-SES which led to the equation for surge acceleration that was presented in Chapter II. The drag characteristics from Ref. 7 for the 3K-SES that is simulated by the DBSIM5D are shown in Fig. 5. The data base program was run using parameters that caused it to exhibit a retarding force corresponding to the curve at sea state four, therefore this curve was used to remodel the RTS5D. The new simplified equation of motion for surge acceleration became

$$\dot{u} = (T_{forw}/m) - (FSD/m) + v \cdot r$$

where FSD is full scale drag which is defined by the curve in Fig. 5. T_{FORW} is the sumation of the four thrusts assumed in this model to be applied at the same point along the center line of the craft.

2. Implementation

FSD was implemented into the RTS5D model by dividing the curve into four separate regions and defining FSD within each region. In region I the retarding force was approximated by a normalized velocity cubed term. In regions II

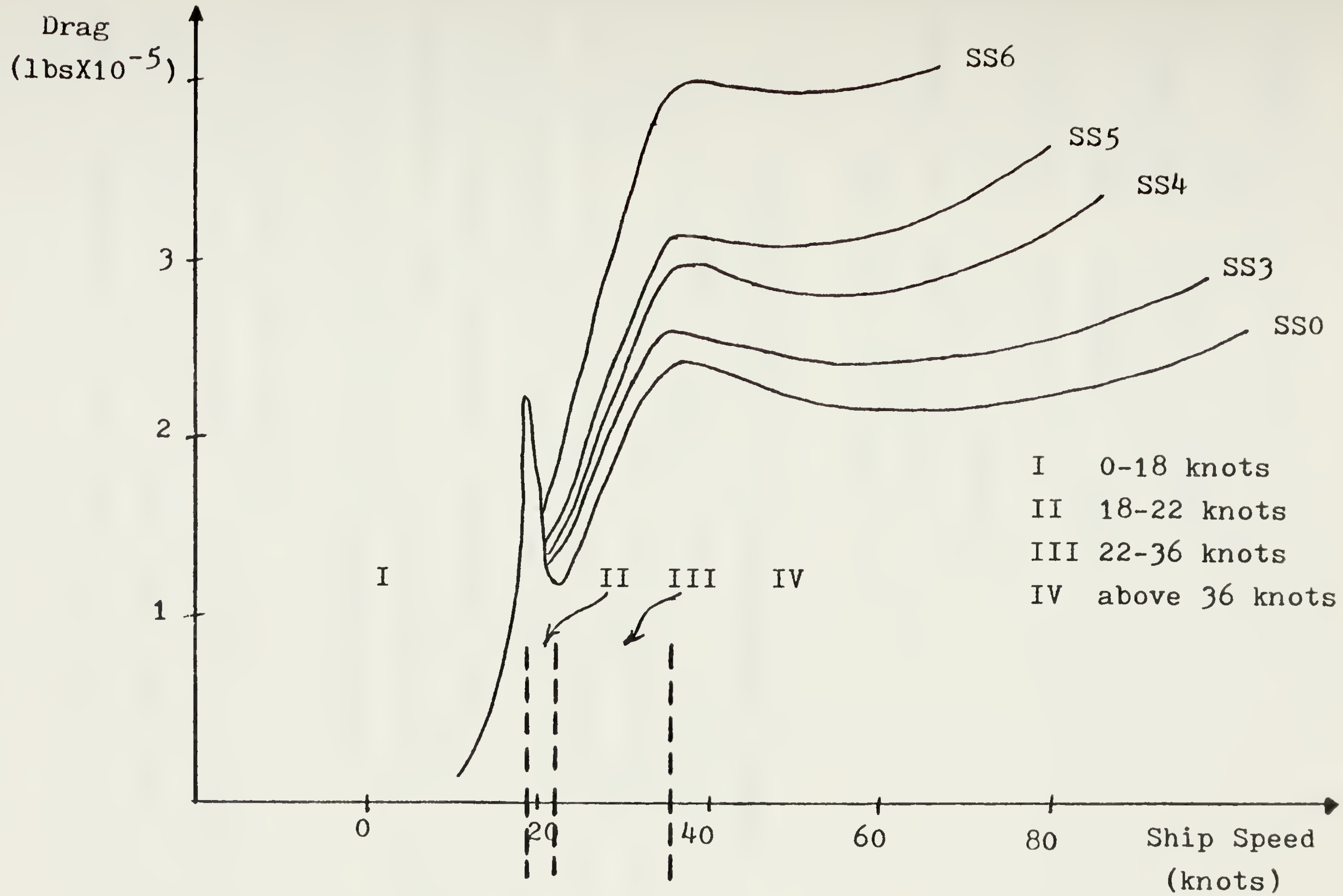


Figure 5

Full Scale Drag Curves

and III straight line approximations were used. In region IV a polynomial curve fit was used. The equations defined for each region are listed below.

Region I - (0-18 knots)

$$FSD = C_{DX1} \left(u / u_{max} \right)^3$$

Region II - (18-22 knots)

$$FSD = CONSTANT_1 - C_{DX5}\, u$$

Region III - (22-36 knots)

$$FSD = C_{DX2}\, u - CONSTANT_2$$

Region IV - (above 36 knots)

$$FSD = C_{DX3}\, u^2 + C_{DX4} / u^{1.5}$$

3. Identification of Surge Drag Coefficients

In region I, C_{DX1} is equal to the first peak value of drag. In regions II and III, C_{DX5} and C_{DX2} are equal to the slopes of the drag curve and the constants were solved for by applying the boundary conditions at the intersections of the adjacent regions. The values of C_{DX3} and C_{DX4} were computed by taking FSD at two different known values of u and solving for C_{DX3} and C_{DX4} simultaneously.

B. LINEARIZED MODEL ANALYSIS

Since the surge model of the RTS5D is extremely nonlinear over the complete speed range, analysis is very complex. Therefore the model was linearized and it's performance

analyzed at various points on the FSD curve. Linearization of the model was accomplished by computing the linear coefficients through partial differention with respect to surge in each of the four region equations and evaluating these terms at various operating point surge velocities (u_o). The linearized model is shown in Fig. 6 and the linear coefficients for the surge drag force FSD are shown below.

Region	
I	$FSD'_{(u_o)} = 3\ C_{DX1}\ (u_o / u_{max})^2$
II	$FSD'_{(u_o)} = -\ C_{DX5}$
III	$FSD'_{(u_o)} = C_{DX2}$
IV	$FSD'_{(u_o)} = 2\ C_{DX3}\ u_o - 1.5\ C_{DX4} / u_o^{2.5}$

A map of the linearized surge directional roots are shown in Fig. 7. The characteristic roots for each operating region are shown coded (see legend). Note that only Region I and Region IV roots are u_o dependent and therefore move as u_o changes. The direction of change along the real axis is noted by the arrows shown. Region II and III roots are fixed in locations and independent of u_o.

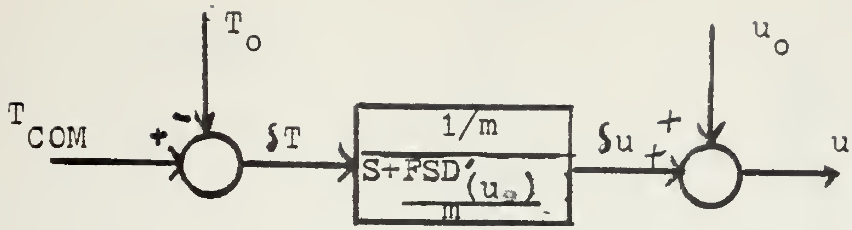


Figure 6

Linearized Surge Model

Region IV of Fig. 7 is of particular interest since the ability to operate in this speed range is a primary consideration in the development of the SES. It can be seen from the map of the roots that within this region, at speeds from 36 to almost 60 knots, poles exist in the RHP. This is a result of the negative drag slope where the retarding force actually decreases with an increase in speed, the magnitude of variation being dependent on sea state. Region II also exhibits this negative drag characteristic but since it is of such a narrow speed range, operation in this region was not considered in this design. Further modeling changes for the other degrees of freedom were based on Region IV response. In Regions I, III and in Region IV above 60 knots, the system will be stable at the speed u_o where FSD is equal to commanded thrust.

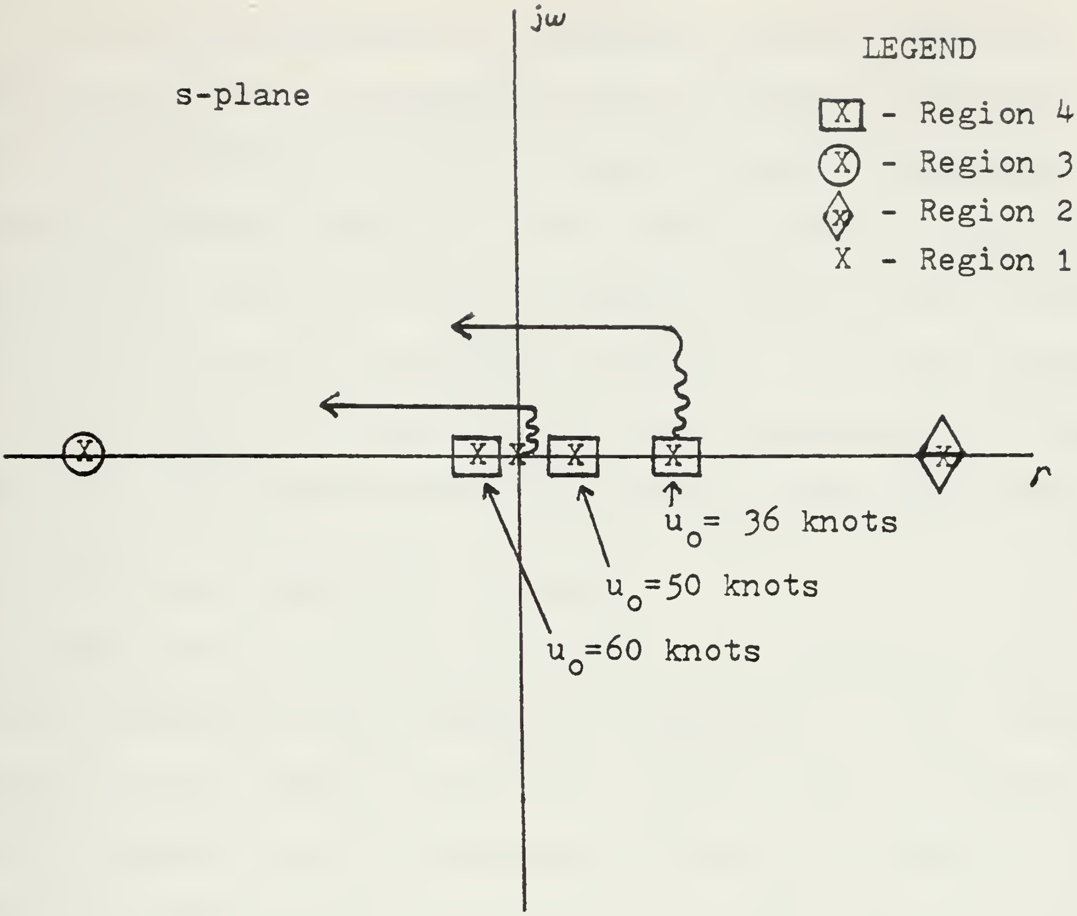


Figure 7
Linearized Surge Root Map

If it were necessary to operate at a speed within the negative drag slope characteristic, it would be necessary to actively modulate the thrust command in order to maintain a nearly constant speed, but this would require a great deal of operator attention since the negative drag slope in Region IV is one of slowly divergent instability. For this reason a closed loop speed controller has been included in the data base model and therefore was also added to the RTS5D model.

C. SIMPLIFIED PROPULSION DYNAMICS

Before adding a speed autopilot to the model one additional refinement was included in the design. Thrust command changes in the RTS5D as reported in Ref. 1 were instantaneously entered into the equations of motion as thrust changes. This was not the case for the data base simulation program. Specific thrust levels are generated by specific water flow rates through water jet nozzles. These flow rates are nearly proportional to the pump speed that produces them. Since the pumps are driven directly by gas turbine engines, a change in thrust occurs with a change in turbine speed. A thrust command translates to a power turbine speed command which as an output has power turbine speed achieved. This output is translated into thrust achieved. It was specified in Ref. 7 that one of the goals in the base line design was to have power turbine speed control response charasteristics which are relatively independent of the power setting. Accordingly, in the design

of the auto pilot of Ref. 7, the simplified propulsion block shown in Fig. 8 was used. The magnitude of the propulsion system time constant, τ was not specified. For the purpose of this analysis τ was chosen to be 1 second.

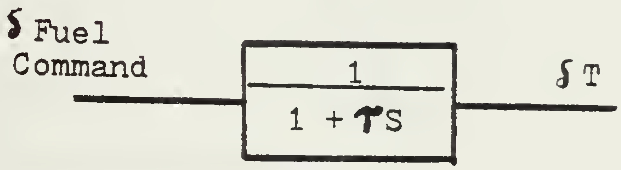


Figure 8

Simplified Propulsion System

D. ANALYSIS OF LINEARIZED MODEL WITH SPEED CONTROLLER

The complete linearized model with speed controller that is included in the model for the purposes of this study is shown in Fig. 9. The saturation effects of fuel control are ignored and the linear representation is shown as K_f which is assumed to be unity. The controller includes proportional plus integral control. The proportional control was included to keep the ship speed error low while the integral control was included in order to zero out long term control error.

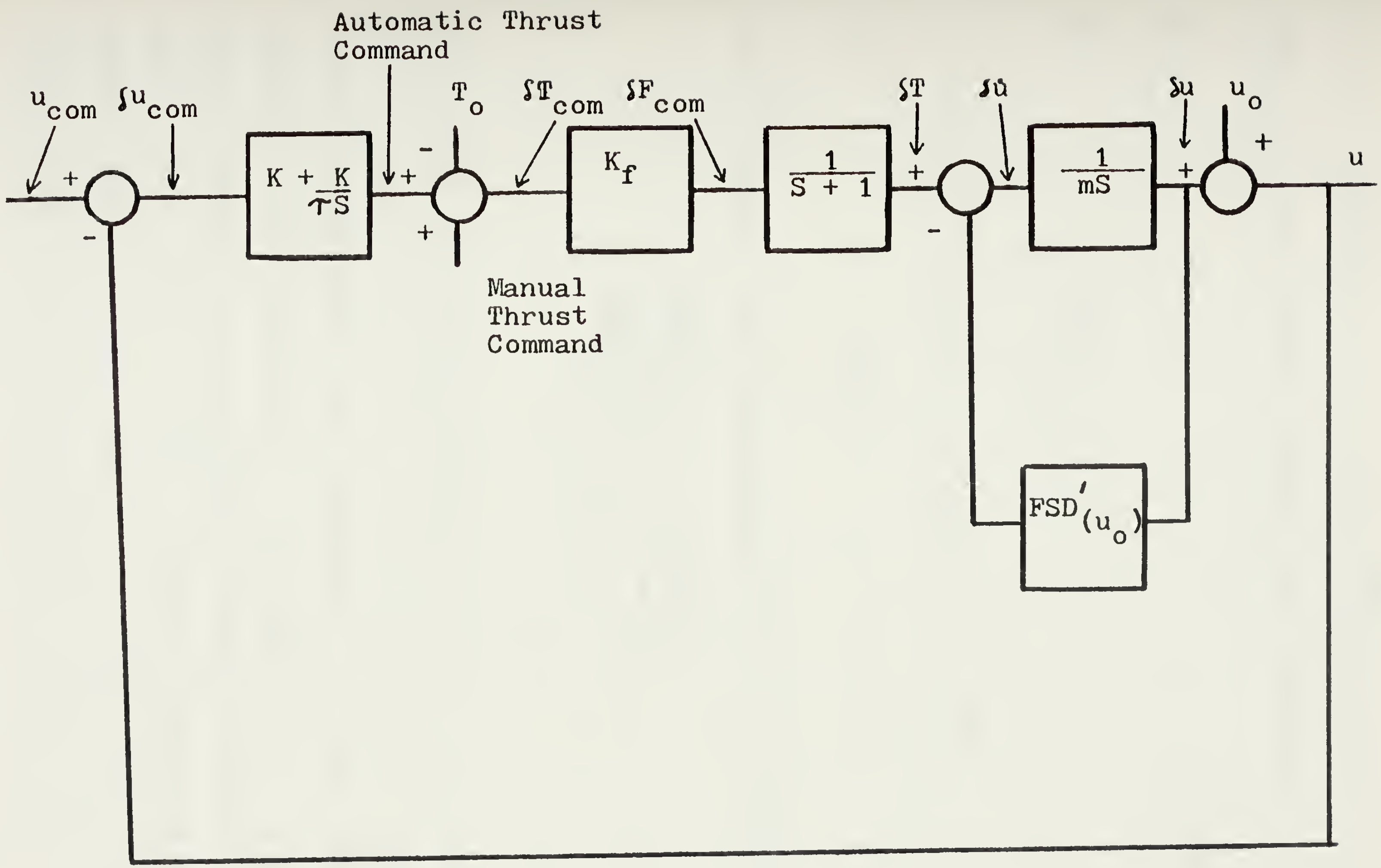


Figure 9

Linearized Surge Model With

Propulsion System and Speed Controller

The characteristic equation for the system in Fig. 9 is

$$S^3 + (1 + FSD'_{(u_o)}/ m) S^2 + 1/m (FSD'_{(u_o)} + K) S + K/ m\tau = 0$$

Using the Routh Hurwitz Criterion for stability

$$\begin{array}{lll}
S^3 & 1 & 1/m\ (FSD'_{(u_o)} + K) \\
S^2 & (1 + FSD'_{(u_o)}/ m) & K/ m\tau \\
S^1 & 1/m\ (1 + FSD'_{(u_o)}/m)\ (FSD'_{(u_o)} + K) - K/m\tau & 0 \\
S^0 & K/ m\tau & 0
\end{array}$$

For Stability K must be greater than 0 and greater than

$$\frac{FSD'^{2}_{(u_o)}\ \tau + m\tau FSD'_{(u_o)}}{FSD'_{(u_o)}\tau + m\tau - m}$$

The values of controller parameters K and τ can be initialized by the operator for either loosely controlled speed or tightly controlled speed for such situations as station keeping, UNREPS, or test operations. For comparison to the data base, these parameters were selected to achieve closest agreement between the values of surge achieved by the RTS5D and DBSIM5D models after the completion of a 360° turn with 50 knot initial surge veloctiy and a thrust effector angle of 15°. This test

condition for the nonlinear models was chosen for two reasons. First, given fuel conservation as a necessity, 50 knots is a likely high speed initial condition because it is close to minimum drag. Second, this initial condition provided an opportunity to test the operation of the speed controller in the negative drag region of FSD since the decrease in speed going into the turn with u_o=50 knots caused increased drag. The values of controller parameters that caused closest agreement between the two nonlinear models, when tested in the linearized model showed instability in the linear sense. This is desirable in a turning mode for the nonlinear model because it allows speed to decrease and the turn to be completed more quickly. Stiff speed control could be achieved by increasing the controller gains, but would normally be used only in straight ahead runs.

Coupling of the equations of motion developed in Chapter II necessitated a repetition of the parameter identification procedure resulting in new values for C_{DY}, ℓw, C_{DP} and C_{DZP}. These values were adjusted for the best curve fit compared to the data base using the linearized model for the RTS5D in a 360° turn using 15° thrust effector angle with initial surge velocity 50 knots. An identical series of tests to those in Chapter III were conducted comparing the new model of the RTS5D to the DBSIM5D. Complete validation results are tabulated in Chapter VI.

E. EFFECTIVE THRUST EFFECTOR ANGLE

As a result of the negative drag characteristics imposed on the model, RTS5D yaw rate (r) was high in the low speed test and low in the high speed test. This opposite error at both ends of the test speed spectrum was reduced by introducing a correction factor to the thrust effector angle proportional to yaw rate

$$Z_{eff} = Z - SLIP \cdot r$$

and adjusting ℓw to increase r. This had the desired effect of reducing error at both ends of the test spectrum since the reduction of the thrust effector angle was greater at the high end.

F. BROACH CONDITION FLAG

As stated in Chapter I, broach response characteristics are not included in this model, therefore a broach condition warning flag was included in the design in order to alert the operator who could then take corrective action to avoid broaching.

Broaching is caused by a water intake being exposed to air and is a function of surge speed, roll angle, pitch angle, drift angle and sea state. For the data base 3K-SES with broach boundaries defined at 10% air by volume, the broaching boundaries in calm water in terms of roll angle, pitch angle and drift angle are shown in Fig. 11 for 40 knots

and Fig. 12 for 60 knots. Reference 6 listed maximum thrust effector angles as a function of surge speed and air flow in order to avoid broaching. This condition is shown plotted in Fig. 10 for fixed plenum air flow rate.

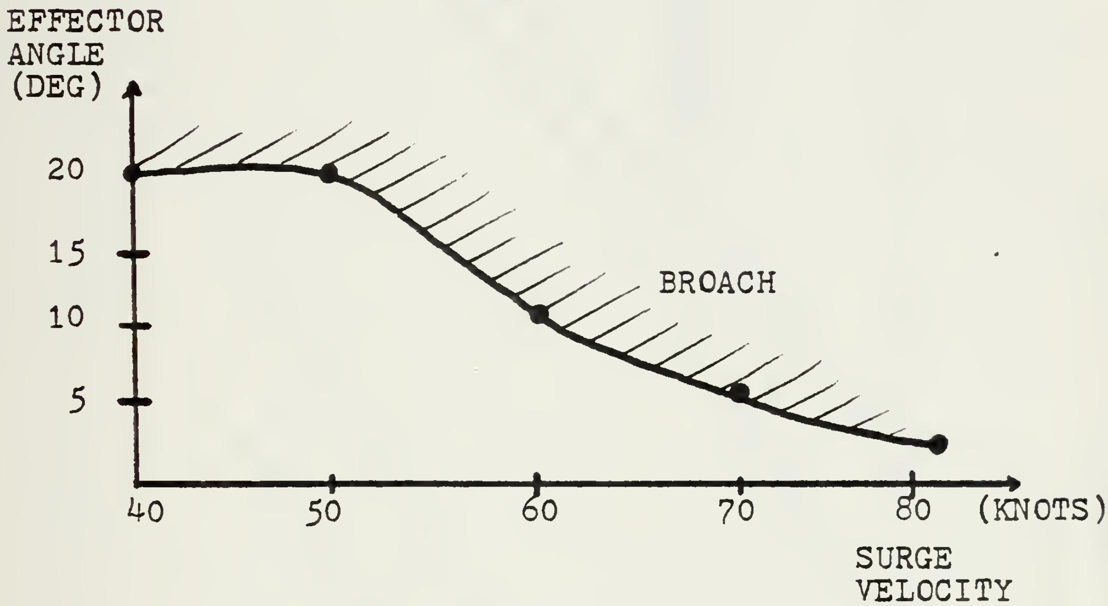


Figure 10

Maximum Thrust Effector Angle As A Function Of Surge Velocity In Order To Avoid Broaching

As stated in Chapter III the model developed in Ref. 1 did not flag the broaching condition. To implement the broach warning flag and display the other new parameters for the modified RTS5D the pilot graphic display described in Ref. 1 was changed to incorporate the necessary data. Figure 13A depicts typical information displayed to the pilot under normal operating conditions. The new information to the

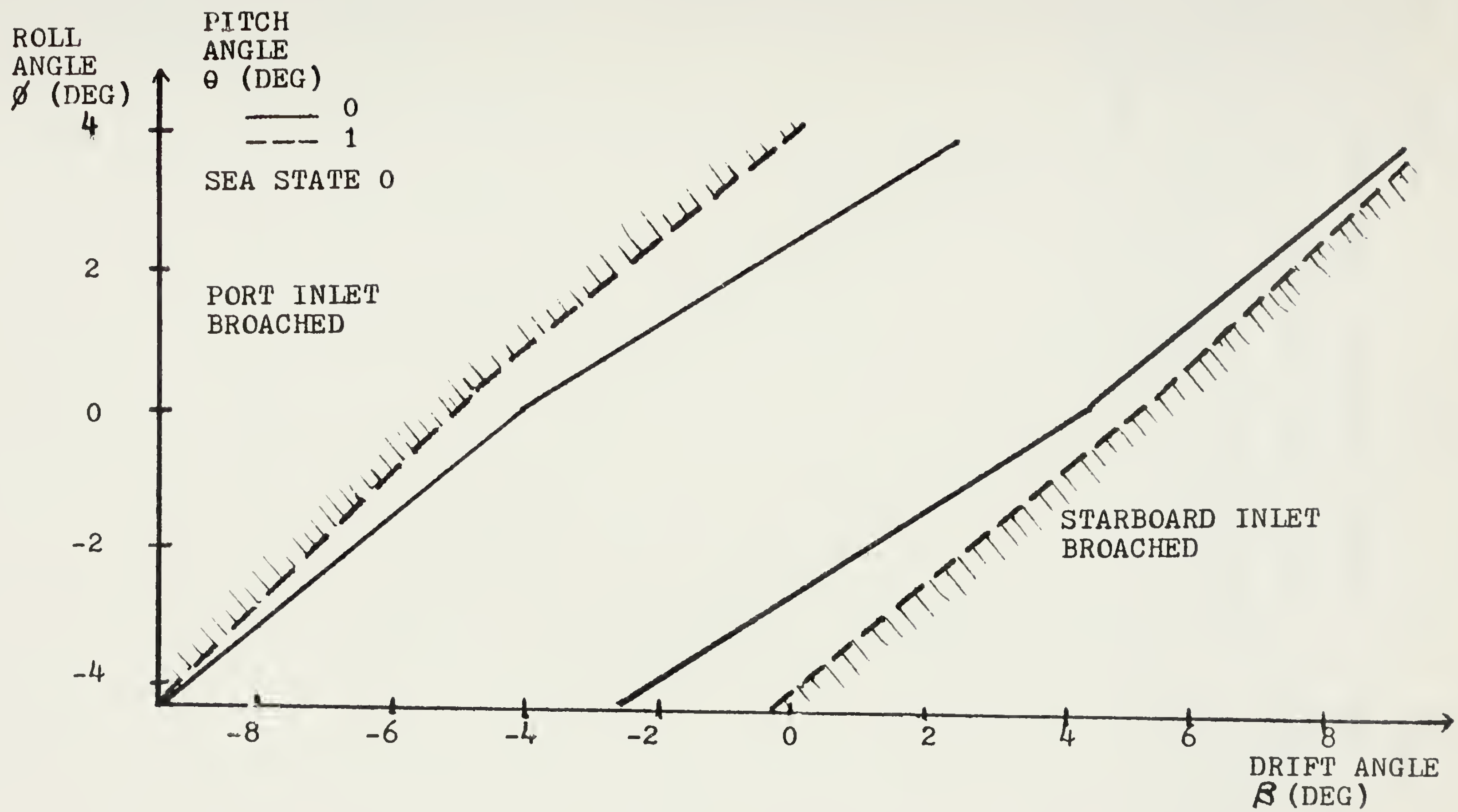


Figure 11

Inlet Broaching Boundaries at 40 Knots

operator shown in the lower right of Fig. 13A is sea state (SEAST), computer assist (CASST, on=1 off=0), controller gains (K,KK) and speed commanded (SPCOM). If the operator exceeds the thrust effector limits (ZMAX) to prevent broaching (see Fig. 10), a flashing warning is displayed in place of the bottom line of information on the screen. This is shown in Fig. 13B.

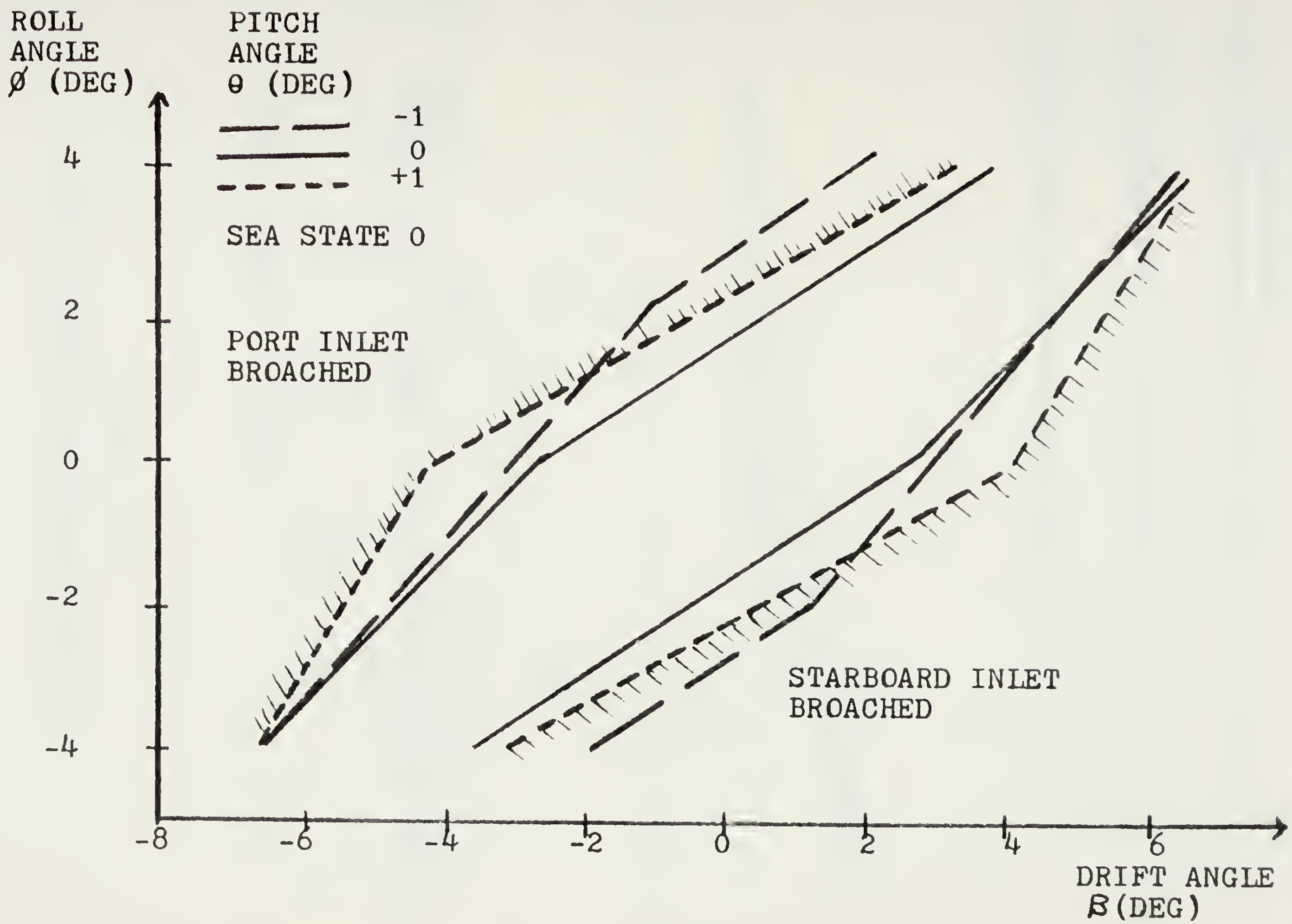
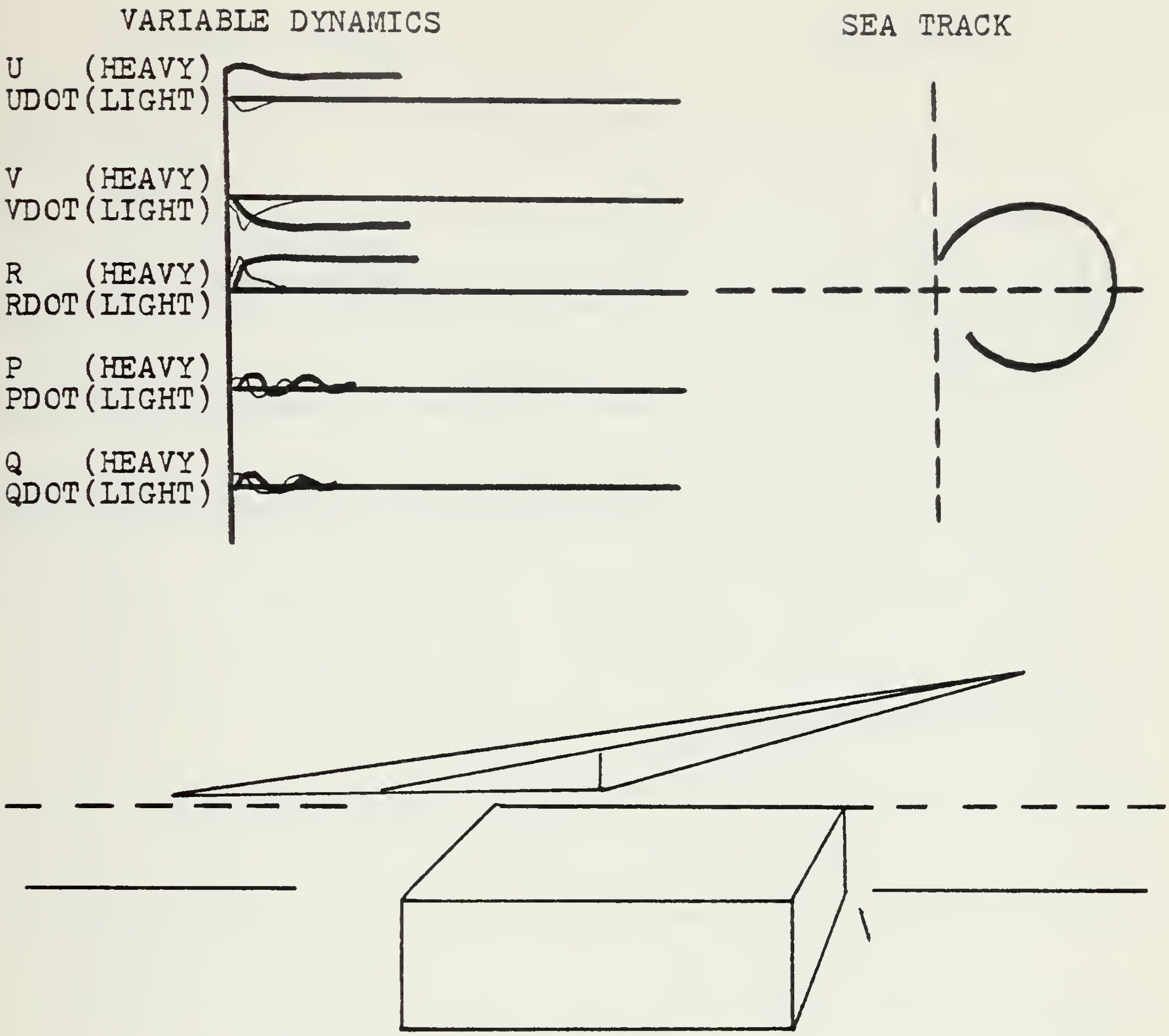


Figure 12

Inlet Broaching Boundaries at 60 Knots



CONTROL INPUTS			TIME		NAVIGATIONAL DATA			
THRUST	RUDDER	DELT	ELAPSED	PITCH	HEADING/RATE	DRIFT	ROLL	SPEED
300000	-.20	.084	22.2	.52	325/2.1	-.28	.8	50.1
T7	T8	T9	T10	SEAST	CASST	K	KK	SPCOM
75000	75000	75000	750000	4	1	2900	3	50

Figure 13A

Pilot Graphic Display (Normal)

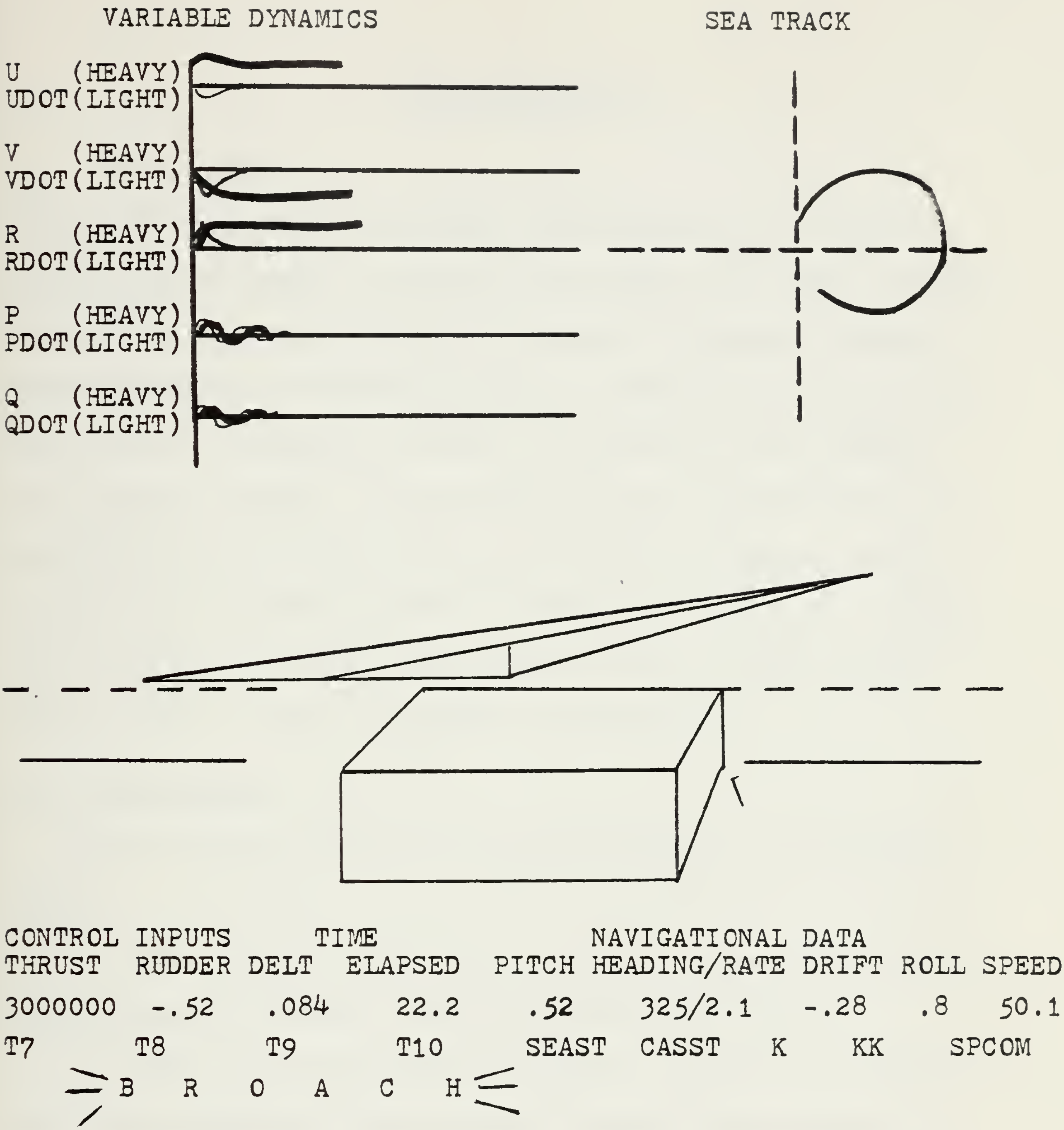


Figure 13B

Pilot Graphic Display (Broach Condition Exists)

V. IMPLEMENTATION

A. INTRODUCTION

Given the existence of a Real Time Simulation of a SES, a decision was made to refine that model using more accurate equations of motion, improved computer iteration time and operator hardware modifications. A complete description of the RTS5D is contained in Ref. 1, therefore only changes are discussed here, and included are the introduction of negative drag in a sea state, propulsion dynamics, speed control and a broach condition warning in software. Hardware modification included changing the thruster control box to make it more accessable to an operator.

B. REQUIREMENTS

The following criteria were required of the remodeled RTS5D.

1) The refined simplified equations of motion would be solved and the results output on a real time basis.

2) The solution would be subject to real time control efforts generated by an operator observing the output.

3) Computer iteration time was required to be less than 100 ms.

A more complex discussion of condition 3) is presented in Appendix A.

C. HARDWARE DESCRIPTION

The hardware arrangement in block diagram form is shown in Fig. 14. The only change to the RTSSD was the redesign and relocation of the thruster console to the pilots seat where it can be more conveniently utilized by the operator. The thruster console uses "linear movement" potentiometers in place of previously used "dial" potentiometers in order to more realistically simulate thrust controls for the SES. The thruster console is shown in Fig. 15.

D. SOFTWARE DESCRIPTION

The remodeled software package was a FORTRAN IV digital program which consisted of a main graphics program and three major subroutines. It was necessary to make a third subroutine out of the equations of motion computation that had previously resided in the main program prior to making any changes to these equations. It was discovered when the first program modifications were attempted that the compiler for the XDS-9300 computer was operating at its limit with the program as it then existed, therefore, before changes to the equations could be implemented, the program structure had to be modified. The failure mode was eliminated from the new model because of initial assumptions that the thrusts would be all applied at the same point. Thrust failures should not be considered until the model includes broaching dynamics which is a major source of thrust loss. The flow

chart for the remodeled RTS5D is shown in Fig. 16. The complete multiplexing algorithm is shown in a flow chart in Fig. 17.

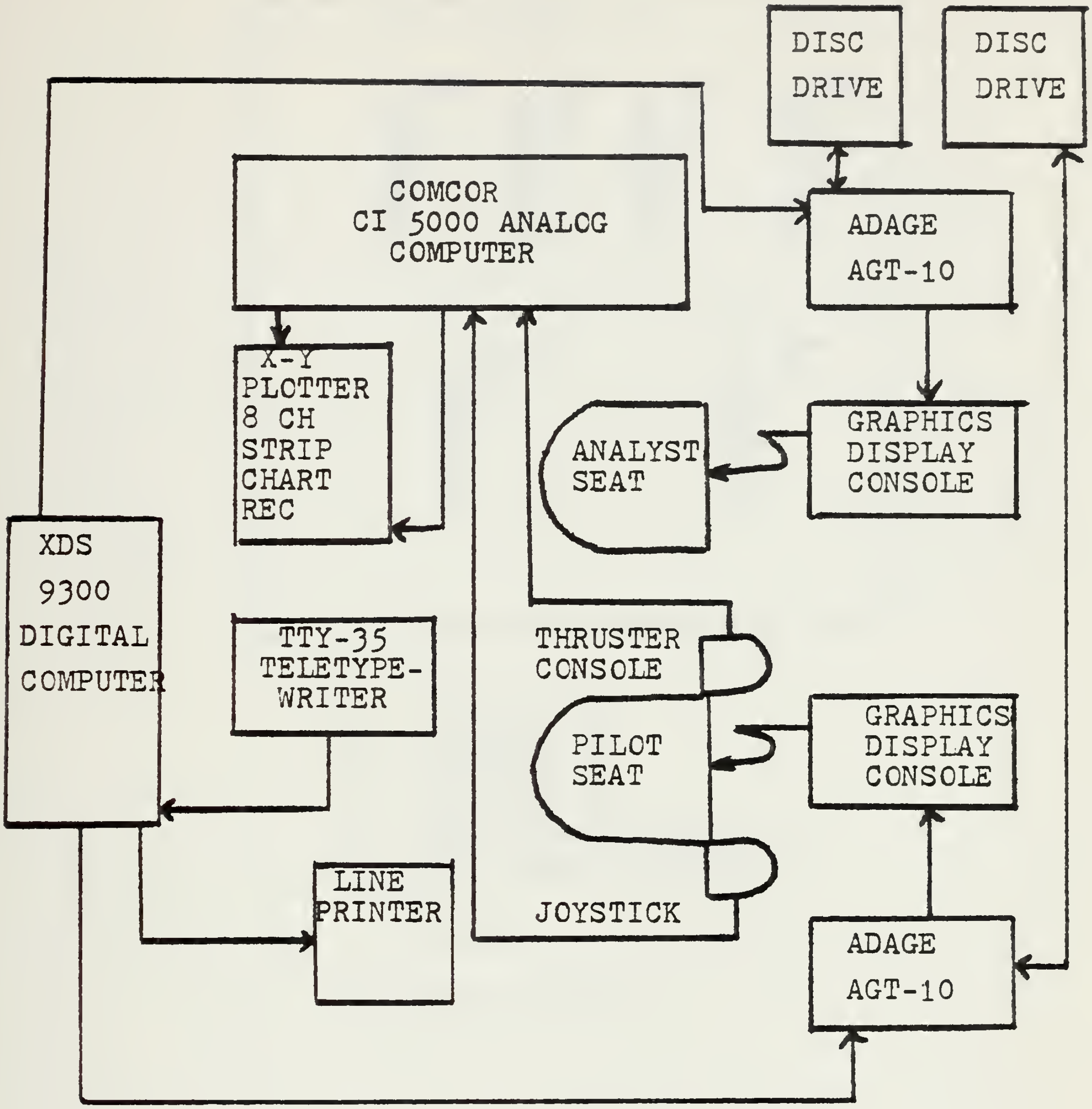
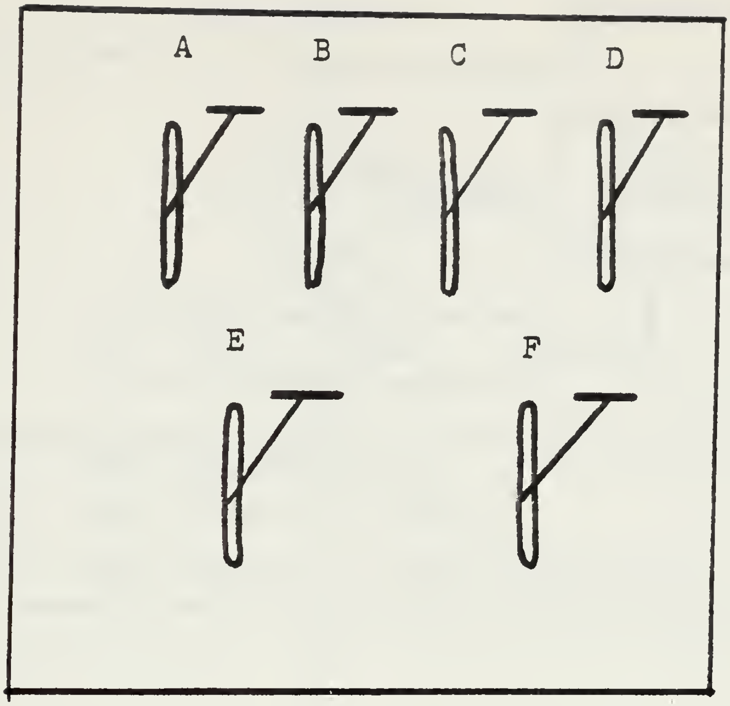


Figure 14

RTS5D Mod Block Diagram



A - Thruster T7
B - Thruster T8
C - Thruster T9
D - Thruster T10
E - Speed Command Control
F - Run, Hold, Restart

Figure 15

Thruster Console

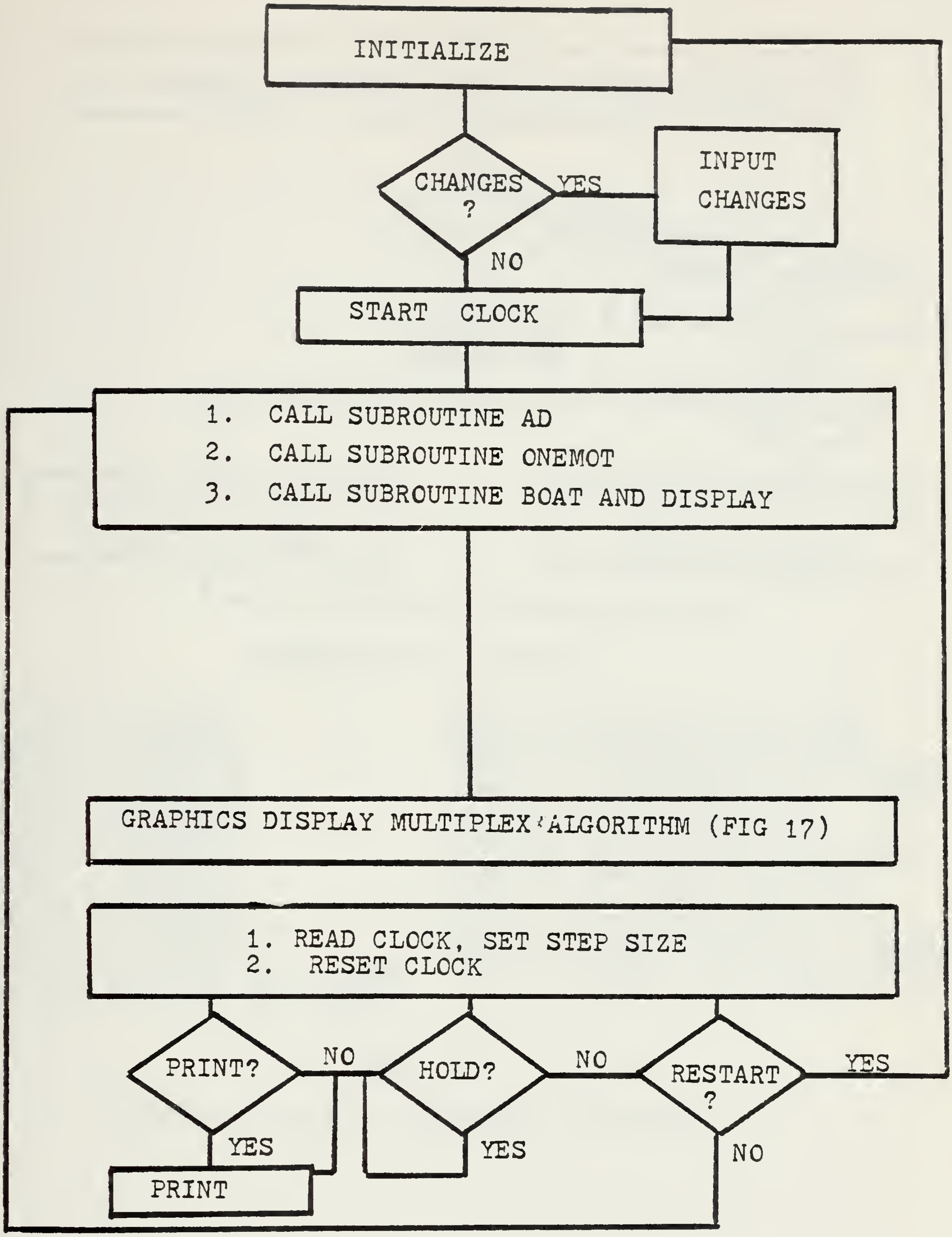
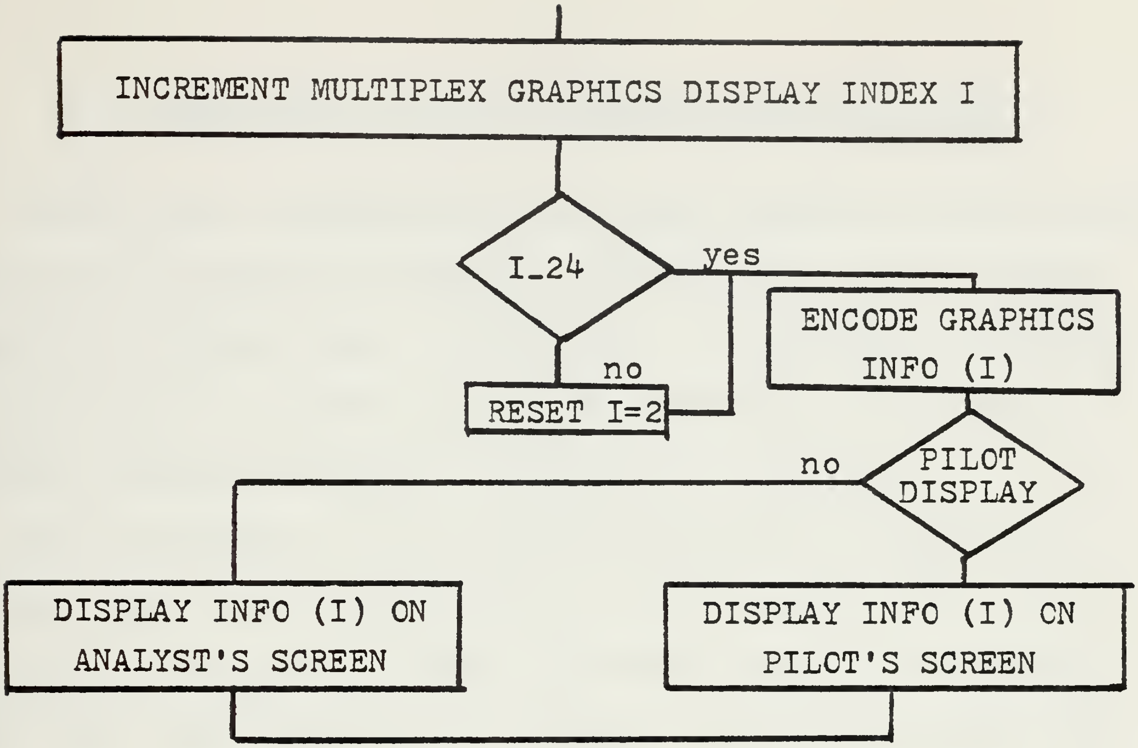


Figure 16

RTS5D Mod Program Flow Chart



DESCRIPTION OF INFO(I)

INFO(1) u DYNAMICS
INFO(2) $\dot{u}$ DYNAMICS
INFO(3) v DYNAMICS
INFO(4) $\dot{v}$ DYNAMICS
INFO(5) r DYNAMICS
INFO(6) $\dot{r}$ DYNAMICS
INFO(7) p DYNAMICS
INFO(8) $\dot{p}$ DYNAMICS
INFO(9) q DYNAMICS
INFO(10) $\dot{q}$ DYNAMICS
INFO(11) POSITION MAX VALUES
INFO(12) POSITION MIN VALUES
INFO(13) VELOCITY MAX VALUES
INFO(14) VELOCITY MIN VALUES
INFO(15) ACCELERATION MAX VALUES
INFO(16) ACCELERATION MIN VALUES
INFO(17) NAVIGATION DATA
INFO(18) EFFECTOR/THRUSTER DATA
INFO(19) SEA TRACK NAV PLOT
INFO(20) SEA TRACK NAV PLOT
INFO(21) PRESENT POSITION VARIABLES
INFO(22) PRESENT VELOCITY VAREABLES
INFO(23) PRESENT ACCELERATION VAR.

Figure 17

RTS5D MOD DISPLAY MULTIPLEX ALGORITHM

VI. RTS5D MODS I AND II RESPONSE CHARACTERISTICS

RTS5D MOD I includes negative drag effects in a sea state, simplified propulsion dynamics, speed control and a broach condition warning flag. RTS5D MOD II is identical to MOD I except that it includes the correction factor for thrust effector angle developed in Chapter IV, Section E.

The validation of the modified RTS5D was accomplished in the same manner as the RTS5D. (See Chapter III). The results are displayed in the following tables. A complete 360° turn comparison of the DBSIM5D, the RTS5D and the modified RTS5D at 50 knots, 15° thruster angle in shown in Fig. 18. Additionally, a comparison of transient response was undertaken and the time of first peak for the variable v, r, ϕ and θ for each model was graphed in Figs. 19-21 for speeds of 40, 50 and 60 knots. Transient response characteristics are an important consideration in the development of a real time model and are discussed in Appendix A. The transient response of the RTS5D models was faster than that of the data base model with the exception of roll angle (ϕ). This impacted on the validity of a real time solution of the equations of motion and the display of the results.

TABLE IV

RTS5D Mod I and Mod II Performance Test at 40 Knots

effector angle	RTS5D Mod I		RTS5D Mod II	
	ϕ in degrees		ϕ	
	1st pk	Qss	1st pk	Qss
5°	.52	.45	.40	.38
10°	.96	.92	.79	.76
	θ in degrees		θ	
	1st pk	Qss	1st pk	Qss
5°	1.21	1.17	1.20	1.18
10°	1.20	1.18	1.20	1.18
	u in ft/sec		u	
	1st pk	Qss	1st pk	Qss
5°	62.56	62.64	62.58	67.65
10°	n/a	61.37	n/a	61.71
	v in ft/sec		v	
	1st pk	Qss	1st pk	Qss
5°	2.62	2.47	2.33	2.27
10°	3.53	3.52	3.23	3.20
	r in degrees/sec		r	
	1st pk	Qss	1st pk	Qss
5°	.69	.59	.57	.54
10°	1.18	1.22	1.04	1.09

TABLE V

RTS5D Mod I and Mod II Performance Test at 50 Knots

effector angle	RTS5D Mod I		RTS5D Mod II	
	ϕ in degrees		ϕ	
	1st pk	Qss	1st pk	Qss
5°	.50	.39	.43	.39
10°	.90	.82	.83	.78
15°	1.31	1.36	1.24	1.19
	θ in degrees		θ	
	1st pk	Qss	1st pk	Qss
5°	1.23	1.19	1.23	1.20
10°	1.24	1.19	1.23	1.20
15°	1.24	1.18	1.23	1.19
	u in ft/sec		u	
	1st pk	Qss	1st pk	Qss
5°	n/a	82.73	n/a	82.98
10°	n/a	78.68	n/a	79.55
15°	n/a	68.30	n/a	72.75
	v in ft/sec		v	
	1st pk	Qss	1st pk	Qss
5°	2.57	2.30	2.39	2.29
10°	3.48	3.33	3.25	3.25
15°	4.13	4.29	3.98	4.00
	r in degrees/sec		r	
	1st pk	Qss	1st pk	Qss
5°	.57	.39	.49	.41
10°	.96	.85	.88	.86
15°	1.31	1.62	1.25	1.41

TABLE VI

RTS5D Mod I and Mod II Performance Test at 60 Knots

effector angle	RTS5D Mod 1		RTS5D Mod II	
	ϕ in degrees		ϕ	
	1st pk	Qss	1st pk	Qss
5°	.53	.39	.46	.41
	θ in degrees		θ	
	1st pk	Qss	1st pk	Qss
5°	1.23	1.20	1.24	1.20
	u		u	
	1st pk	Qss	1st pk	Qss
5°	n/a	100.21	n/a	99.59
	v		v	
	1st pk	Qss	1st pk	Qss
5°	2.64	2.29	2.49	2.35
	r		r	
	1st pk	Qss	1st pk	Qss
5°	.52	.32	.46	.36

u_o = 50 knots

z = 15° (effector angle)

10 second intervals

o RTS5D
Δ DBSIM5D
• RTS5D MOD I
x RTS5D MOD II

Figure 18

360° Turn Comparison of Four Models

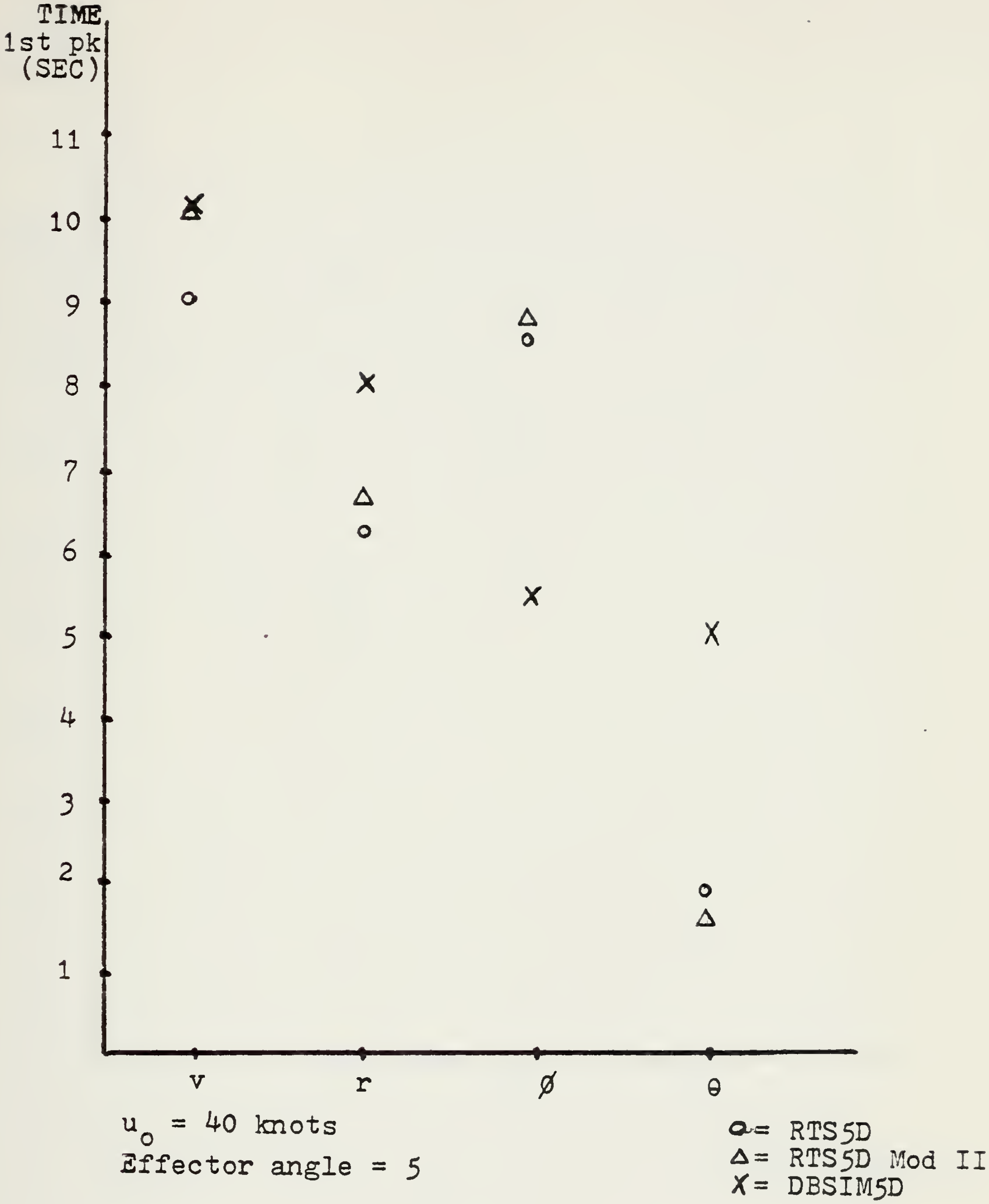


Figure 19

Response Time Comparison at 40 Knots

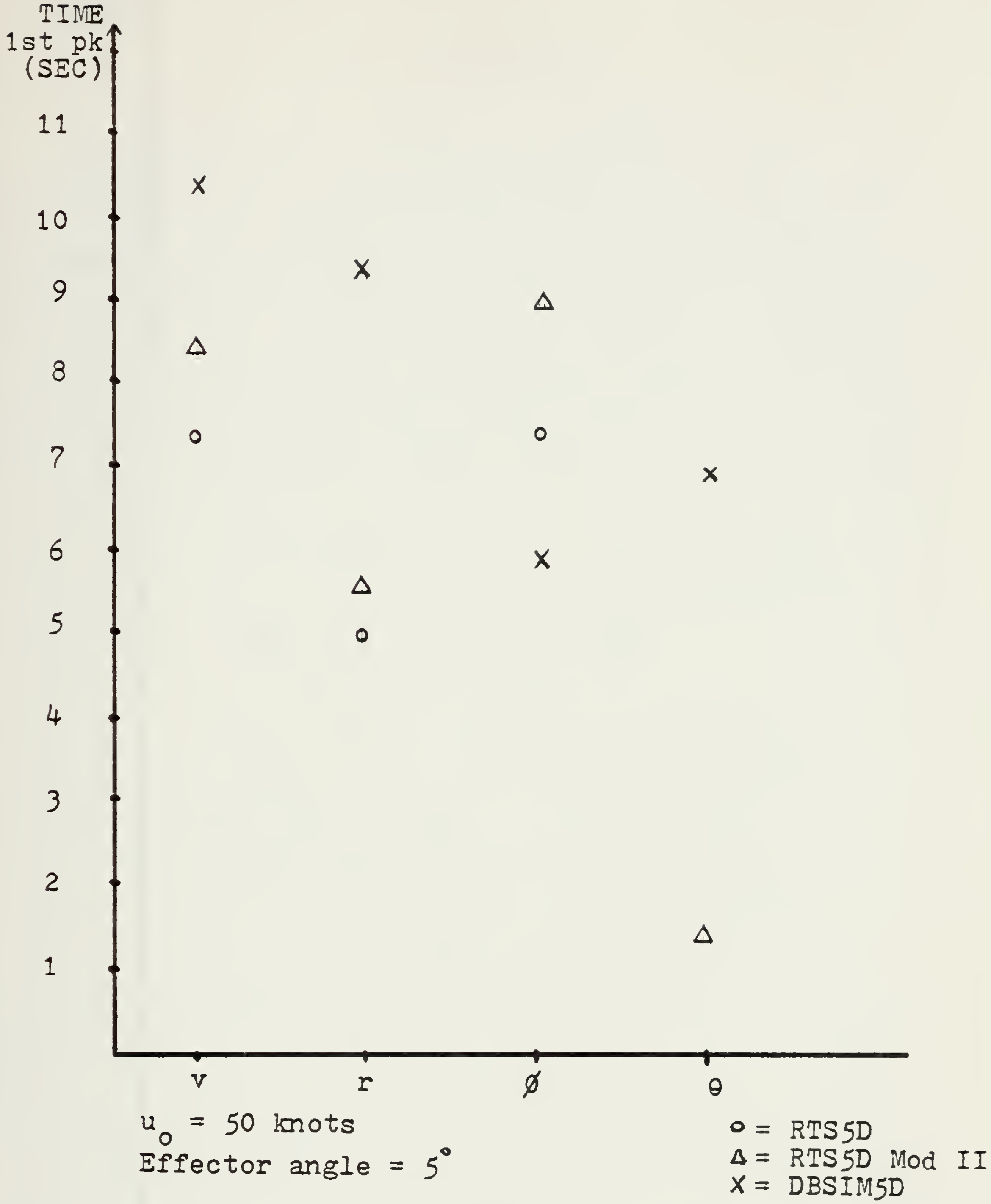
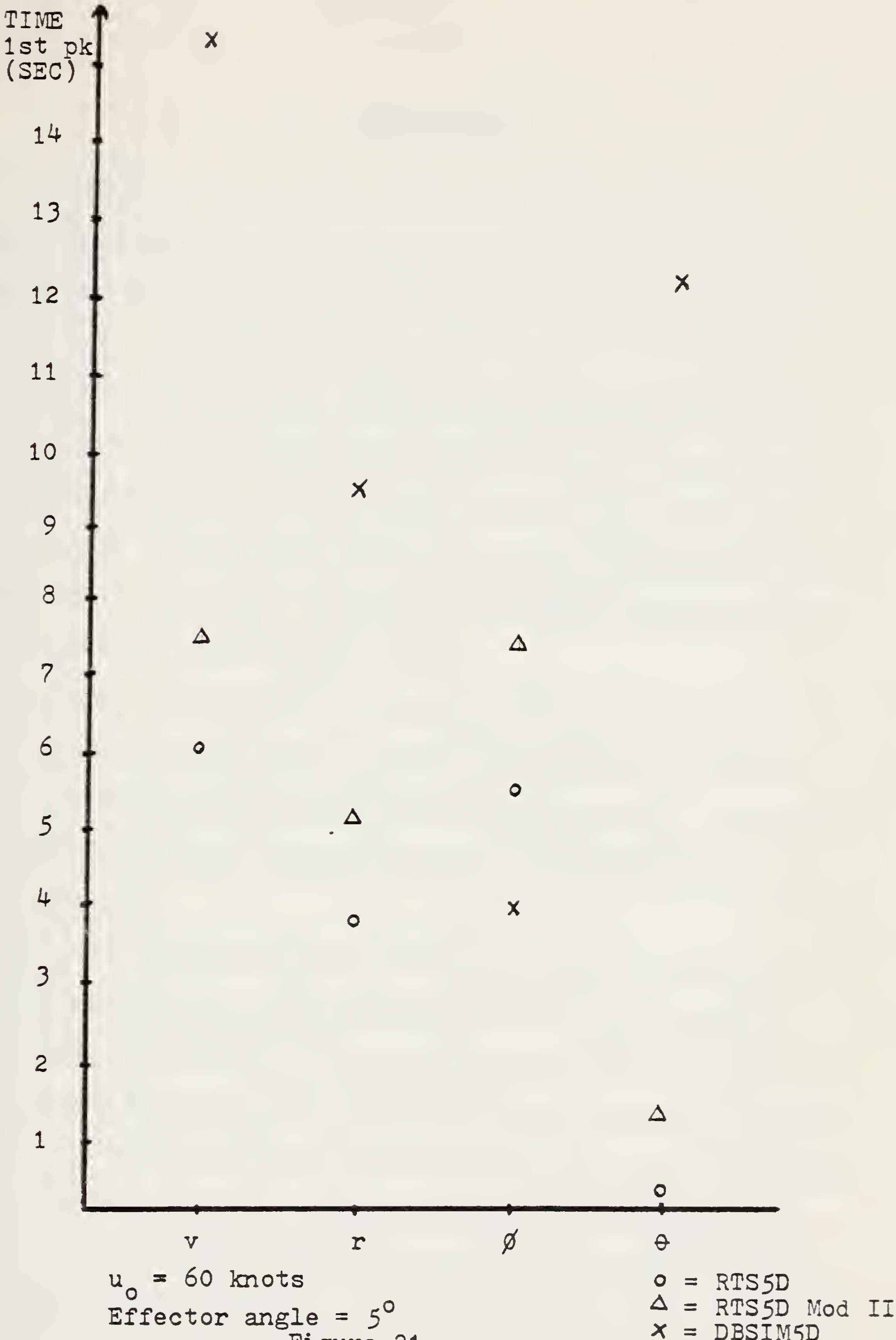


Figure 20

Response Time Comparison at 50 Knots



u_o = 60 knots

Effector angle = 5^o

Figure 21

Response Time Comparison at 60 Knots

VII. CONCLUSIONS

The RTS5D has been shown in Section VI to be a viable input/output model of a 3K-SES by closely matching its output characteristics for a given input to those of the data base model.

The refinement of the right side of the simplified equations of motion, especially the inclusion of the negative drag characteristic in a sea state has made the model more accurate over a wide speed range and gives the operator, upon program initialization, the ability to choose different drag characteristics caused by sea state.

The model was tested in a negative drag region with automatic speed control and simplified propulsion dynamics and it was demonstrated that by properly adjusting controller gains that the output characteristics of the DBSIM5D were closely matched.

Previously conducted thrust failure analysis with man generated corrective action has been shown to be invalid until broach dynamics are included into the model. Previous tests allowed maximum thrust effector angle correction regardless of this consideration. The modified version of the RTS5D does not include complete broach dynamics but does warn the operator if he should exceed a maximum thrust effector angle for a given forward speed.

The 3K-SES may be operated in the negative drag slope speed region approximately (36-58) knots without the aid of a speed controller but would require constant manual modulation of the thrusters. Since minimum manning is one of the criteria in 3K-SES design, a speed controller is necessary to free operator attention to other duties when operating in this speed range.

Another consideration of the implementation of the non-linear drag curves is that for a given sea state, two values of minimum drag can be determined, one in Region IV and one between Region II and Region III. Knowledge that drag can be lower in a particular operating region for fuel conservation is of prime importance.

It is recognized that a time delay exists in the solution of the equations of motion and a time delay exists in the graphical display of the solutions to the operator. Real time interaction between the man in the loop and the computer is achieved by setting the integration step size equal to the computer loop iteration time. Real time considerations are further discussed in Appendix A.

VIII. RECOMMENDATIONS

The following recommendations are made concerning future improvements in the RTS5D model.

1. Further refinement of the right hand side of the equations of motion is necessary in order to more closely match the characteristics of the data base model.

2. Implement software broach dynamics in the model with thrust applied differentially at different lever arms (see Fig. 3) so that thrust failures and effectiveness of corrective action may be evaluated.

3. Rewrite the program such that the AGT-10's would provide asyncronous parallel graphics (existing equipment) or implement the model on a more modern system which uses parallel processing and improved graphics.

4. Introduce the sixth degree of freedom (heave) as a parallel operation. Since heave time constant is much faster than the rest of the system dynamics, real time heave analysis could be conducted simultaneously if cross coupling is neglected.

5. The system is designed for man in the loop interface, so further operator testing is recommended as a valuable source of information for future refinement of the model.

APPENDIX A

REAL TIME ANALYSIS

T. S. Nelson, in Ref. 1, accomplished real time solution of SES equations of motion with the use of a 1000 Hz clock to measure computer iteration time and by using that time as an integration step size for rectangular integration to solve the simplified dynamic equations. He concluded, by comparing the solutions of these equations to that obtained by using a Runga Kutta variable integration step size, that the computer iteration loop time must stay below 100 ms in order to maintain at least 3% accuracy of solution. He further developed a multiplexing algorithm in order to output these solutions to an operator in the form of graphics on a real time basis. This model can be invalidated as a real time model in two ways. First, in order to have real time, the integration step size used must be equal to the computer iteration time and in this model, because of the multiplexing algorithm, each loop time is different. The integration step size, dt determined by measuring loop N is used to solve the equations in loop N+1, which has a different loop time, therefore the solutions are not exactly real time. Secondly, the multiplexing scheme makes the model into a sampled data system with sample time (Ts) approximately 2 seconds. Even if the first condition did not apply and real time solutions to the equations could be assumed, then the output to the

operator would be sampled real time data. Given the fastest time constant for the RTS5D simplified equations (approximately 1.3 seconds for pitch rotational motion), it can be seen that the Nyquist Criteria is clearly violated. Yet even with these obvious limitations, it has been demonstrated that the RTS5D can closely reproduce the output characteristics of the DBSIM5D and therefore is a useful model. Surge speed as well as sway, yaw and roll (Figs. 21-23) have much longer effective time constants. It is recognized that there is a delay between the real time of any event and the observation of that event, therefore "Real Time" as used in this report is a matter of degree, or more specifically how much delay time between solution and display can be tolerated. Given the scale of graphics presented to the operator in this model and the speed with which the SES responds, the RTS5D is an acceptable model, however it would be desirable to reduce the sample time T_s and make the integration step size equal to the computer loop iteration time for each iteration.

An algorithm to set the integration step size equal to the computer iteration time of the current loop was implemented as a software design change to the RTS5D model. This was accomplished by defining the integration step size as an array DELT(N) with N equal to the total number of iterations necessary to display all of the graphical information to the operator. Each loop iteration time was measured and

stored in this array during the first pass through the multiplexing algorithm. DELT(N) was initialized at zero so that time would not increment and integration would not be performed during the first pass through the multiplexing algorithm. On subsequent passes through the multiplexing algorithm the integration step size was equal to the loop time in which it was used.

This timing algorithm when tested in the RTS5D model was found to accurately represent real time but the deviation from previously used integration step size values caused instability in both the pitch and roll dynamics. This needs further investigation.

Sturgeon, in Ref. 8, demonstrated the value of using parallel array processors to reduce computational time in real time simulations. Such a model could be designed for the RTS5D using the existing equipment. The AGT-10's which are used for displaying the output can interface with the XDS-9300 main computer in two ways. The multiplexing scheme of the RTS5D uses a "hand shaking" method with a GRAPHO or TEXTO subroutine call in each loop. It is precisely the inclusion of graphics in each loop that causes problems for real time presentation. The solution of the equations requires only approximately 35 ms. Each GRAPHO or TEXTO call, with its associated "hand shaking" time, makes up the rest of the loop time, approximately 40 ms more. It can be seen that by introducing two such calls in one loop that the 100 ms time restriction would be exceeded.

An improvement to this method would be to use the second interface between the AGT-10's and the XDS-9300 which is that the AGT-10's have a direct connection to some memory locations in XDS-9300. By solving the dynamics for the SES and storing the data in these memory addresses in the form of arrays, the AGT-10's could be made to operate asyncronously in parallel as array processors by using a process known as "cycle stealing" to fetch the data from the XDS-9300 and to display this on the screen for the operator. This process would have the advantage of reducing and making more uniform the computer iteration time used as the step size for the rectangular integration in the solution of the SES equations of motion. The sample time would also be reduced because all of the time wasted in the "hand shaking" is eliminated. It's drawback is that the output is not truely real time because of the asyncronous operation of the AGT-10's. However, the reduced sampled data time and computer iteration time would make it a closer approximation of real time than presently exists.

APPENDIX B

RTS5D MODIFIED PROGRAM NOMENCLATURE

A11	A/D trunk 500 voltage
A21	A/D trunk 501 voltage
A22	Added mass coefficient in yaw moment equation
A31	A/D trunk 502 voltage
A33	Added mass coefficient in roll moment equation
A34	Added mass coefficient in pitch moment equation
A41	A/D trunk 503 voltage
ABSYD	Y_o
ABSYOD	$\dot{Y}_o$
AD	array of A/D lines
A1COA	speed line index counter
AM	mass
APITCH	pitch angle in degrees
APRINT	# iterations between print execute commands
ARR	Real IARR (1)
ARATE	turn rate in degrees
AROLL	roll angle
ASTOP	performance index J_1 limit
AXST	initial condition X-coordinate on sea track
AYST	initial condition Y-coordinate on sea track
BETA	drift angle
BOAT	subroutine to generate perspective of SES

CASST	automatic speed control mode indicator
CDP	bow real forces lumped coefficient
$CDX_{(i)}$	drag forces lumped coefficients
CDY	sway forces lumped coefficient
CDZP	sidewall rolling moment lumped ceofficient
DAL	digital to analog call
DELT	iteration loop time
DELTA	commanded iteration loop time
DFCOM	total change in fuel command from speed comtroller
DFCOM 1	change in fuel command due to proportional speed controller
DFCOM 2	change in fuel command due to proportional speed controller
DTIME	v time
DTIMPLT	scaled value for v
DUCOM	difference between ucom and u
DXPLOT	same as DTIMPLT
DYPLOT	scaled value for v
DYREPET	restart value for v
EFLG	error flag for TEXTO call
ETIME	r time
EYREPET	restart value for r
EXPLOT	scaled r time
EYPLOT	scaled r
F1	stern buoyancy force
F2	bow seal pitch force
F3	bow buoyance force

FCOM	fuel command
FSP	port sidewall buoyancy force
FSS	starboard sidewall buoyancy force
FTIME	$\dot{v}$ time
FTIMPLT	scaled $\dot{v}$ time
FXPLOT	same as FTIMPLT
FYPLOT	scaled $\dot{v}$
FYREPET	restart value for $\dot{v}$
G	gravitational acceleration
GRAPHO	subroutine to project image (non alpha-numeric)
GTIME	$\dot{R}$ time
GTIMPLT	scaled $\dot{R}$ time
GXPLOT	same asGTIMPLT
GYPLOT	scaled $\dot{R}$
GYREPET	restart $\dot{R}$ value
$H_{(i)}$	dynamics maximum and minimum array
HEAD	craft heading
HOLD	subroutine to freeze display/program
HTIME	$\dot{u}$ time
HTIMPLT	scaled $\dot{u}$ time
HYREPET	analog restart y coordinate on sea track
HYSESX	analog y position of craft
HYSESY	analog x position of craft
IARR(1)	digitized mode selector
IARR(2)	digitized effector angle
IARR(3)	digitized T7

IARR(4)	digitized T8
IARR(5)	digitized T9
IARR(6)	digitized T10
IARR(7)	digitized surge velocity command
ICO	multiplex graphics index
ICOA	speed line index
ICRAFT	vehicle designator
IER	error flag
IIPR	time history limit
IIPRD	time history limit
IIPRE	time history limit
IIPRF	time history limit
IIPRG	time history limit
IIPRH	time history limit
IIPRC	time history limit
IIPRP	time history limit
IIPRQ	time history limit
IIPRR	time history limit
IIPRS	time history limit
IIX	print counter
IJ	craft perspective counter
IJA	craft perspective counter
IJAD	time history counter
IJAE	time history counter
IJAF	time history counter
IJAG	time history counter
IJAH	time history counter

IJAO	time history counter
IJAP	time history counter
IJAQ	time history counter
IJAR	time history counter
IJAS	time history counter
IJD	time history counter
IJE	time history counter
IJF	time history counter
IJG	time history counter
IJH	time history counter
IJO	time history counter
IJP	time history counter
IJQ	time history counter
IJR	time history counter
IJS	time history counter
IL	graphics console display array
ILA1	graphics console display array
ILA2	graphics console display array
ILA3	graphics console display array
ILA4	graphics console display array
ILA5	graphics console display array
ILA6	graphics console display array
ILA7	graphics console display array
ILA8	graphics console display array
ILA9	graphics console display array
ILA10	graphics console display array

ILA11	graphics console display array
ILA12	graphics console display array
ILA13	graphics console display array
ILA14	graphics console display array
ILA15	graphics console display array
ILA16	graphics console display array
ILA17	graphics console display array
ILA18	graphics console display array
ILA19	graphics console display array
ILA20	graphics console display array
ILA21	graphics console display array
ILA22	graphics console display array
ILA23	graphics console display array
ILA24	graphics console display array
ILA25	graphics console display array
ILA26	graphics console display array
ILA27	graphics console display array
ILA28	graphics console display array
ILA29	graphics console display array
ILAP	graphics console display array
IPACK	subroutine to load display arrays
IPLOT	graphics console display array
IPR	time history counter limit
IPRD	time history counter limit

IPRE	time history counter limit
IPRF	time history counter limit
IPRG	time history counter limit
IPRH	time history counter limit
IPRO	time history counter limit
IPRP	time history counter limit
IPRQ	time history counter limit
IPRR	time history counter limit
IPRS	time history counter limit
ISWA	display array
ITEXT	subroutine to display alpha-numeric data
IVEIWA	craft perspective/north road array
IVIEWAA	sea track array
IVIEWB	sea track axis array
IVIEWC	u time history array
IVIEWD	v time history array
IVIEWE	r time history array
IVIEWF	$\dot{v}$ time history array
IVIEWG	$\dot{r}$ time history array
IVIEWH	$\dot{u}$ time history array
IVIEWO	$\dot{p}$ time history array
IVIEWP	$\dot{q}$ time history array
IVIEWQ	p time history array
IVIEWR	q time history array
IVIEWS	analog sea track array
IVIEWZ	time history axis array

IX	x axis moment of inertia
J	time history counter
JD	time history counter
JE	time history counter
JF	time histroy counter
JG	time history counter
JH	time history counter
JO	time history counter
JP	time history counter
JQ	time history counter
JR	time history counter
JS	time history counter
K	proportional speed controller gain
KK	intregal speed controller gain
L3	lever arm for buoyancy force
LD	draft of sidewall
LP	side thrust lever arm
LZ	lever arm for sway drag force
N	clock interrupt count
NAD	AD trunk line array
ONEMOT	subroutine to calculate SES dynamics
OO	lever arm for thrust side component
OTIME	$\dot{p}$ time
OTIMPLT	scaled $\dot{p}$ time
OXPLOT	same as OTIMPLT
OYPLOT	scaled $\dot{p}$
OYREPET	restart $\dot{p}$ value

P	roll rate
PB	plenum pressure
PDOT	roll acceleration
PDOTM	max roll acceleration
PHI	roll angle
PMAX	max roll rate
PSI	craft heading
PTIME	$\dot{q}$ time
PTIMPLT	scaled $\dot{q}$ time
PXPLOT	same as PTIMPLT
PYPLOT	scaled $\dot{q}$
PYREPET	restart $\dot{q}$ value
Q	pitch rate
QDOT	pitch acceleration
QDOTM	max $\dot{q}$
QMAX	max q
QTIME	p time
QTIMPLT	scaled p time
QXPLOT	same as QTIMPLT
QYPLOT	scaled p value
QYREPET	restart p value
R	craft turn rate r
RDOT	$\dot{r}$
RDOTM	max $\dot{r}$
READCLOCK	subroutine to sample real time clock

RESET	subroutine to reset analog computer
RHO	density of water
RMAX	maximum yaw rate of craft
RTIME	q rate
RTIMPLT	scaled q time
RX	I_X
RXPLCT	same as RTIMPLT
RY	I_Y
RYPLCT	scaled q value
RYREPET	restart q value
RZ	I_Z
SEAST	sea state
SESX	scaled X_o
SESY	scaled Y_o
SF1	scale factor in time history arrays
SF14	scale factor in time history arrays
SLIP	Beta influence coefficient
SPDLIN	speedline lower screen limit
SPEED	Craft's velocity in knots
STARTCLOCK	subroutine to start clock
STIME	analog time
STIMPLT	scaled analog time
SXPLCT	HYSESX
SYPLCT	HYSESY
SXREPET	restart value for HYSESX
SYREPET	restart value for HYSESY

T	total thrust
T10	thruster number 4
T10MAX	max T10 value
T7	thruster number 1
T7MAS	max T7 value
T8	thruster number 2
T8MAX	max value of T8
T9	thruster number 3
T9MAX	max value of T9
TEXTO	subroutine to display alpha-numeric data
TDOT	$\dot{T}$
THETA	pitch angle
TIM	u time
TIME	time
TIMPLT	scaled TIM
TINT	real clock interrupt frequency
TITLE	program name display array
TITLEO	program name display array
TITLE1	program name display array
TM	manual thrust command
TMAX	max T
TSIDE	sum of side thrust components
TYAW	sum of yaw moment
U	forward velocity (surge)
UCOM	forward velocity (surge) command

UDOT	$\dot{u}$
UDOTM	max $\dot{u}$
UMAX	max u
UMUL	scale factor in speedline generation
V	side velocity (sway)
VCD	thruster console pot array
VDOT	$\dot{v}$
VDOTM	max $\dot{v}$
VMAX	max v
VS	total velocity
W	heave velocity
WDOT	heave acceleration
WE	width of bow seal
WRITECLOCK	subroutine to assign clock interrupts to variable
WW	lever arm for sway drag forces
X	Y_o
XO	X_o
XODOT	$\dot{X}_o$
XDX	boat perspective scale factor
XGX	boat perspective scale factor
XPLOT	display array
XREPET	reset value for SESX
XST	scale factor for analog plot
YO	Y_o
YODOT	$\dot{Y}_o$

YHIGH	point D vertical position constant
YPLOT	display array
YREPET	reset value for SESY
YST	scale factor for analog plot
Z	operational effector angle
Z2	upper effector angle limitation
Z3	lower effector angle limitation
Z4	zero angle upper limit
Z5	zero angle lower limit
ZAB	scale factor for Z10
ZBRC	effector broach limit angle
ZI	modified Z
ZZZZ	fixed step effector angle input

APPENDIX C

RTS5D MODIFIED COMPUTER PROGRAM LISTING

```
      DIMENSION ITEXT(29),ILAF(24),IVIEWA(75),IVIEWB(70),ISWA(40),
     CIL(29),TITLE(24),IARR(7),ILA1(24),ILA2(24),IVIEWC(205),IVIEWZ(70),
     CILA4(24),ILA5(24),ILA6(24),ILA7(24),ILA8(24),IPLOT(29),ILA3(24),
     CTITLEC(24),TITLE1(24),IVIEWD(205),IVIEWE(205),IVIEWF(205),
     CIVIEWG(205),IVIEWH(205),ILA10(24),ILA11(24),ILA12(24),ILA13(24),
     CILA14(24),ILA15(24),ILA16(24),IVIEWAA(500),IVIEWJ(205),
     CIVIEWP(205),IVIEWQ(205),IVIEWR(205),IVIEWS(205),
     CILA17(25),ILA18(24),ILA19(24),ILA20(24),ILA21(24),ILA22(24),
     CILA23(24),ILA24(24),ILA25(24),ILA26(24),ILA27(24),ILA28(24),
     CILA29(24),F(50),IIPR(20),IPR(20),J(20),IJ(20),IJA(20),XREPET(20),
     CYREPET(20),TIM(20),TIMPLT(20),XPLOT(20),YPLOT(20)
      INTEGER TITLE,TITLEC
      EQUIVALENCE (TITLEO,TITLE1)
      EQUIVALENCE (ILAI,ILA1),(ILAI,ILA2),(ILAI,ILA3),(ILAI,ILA4),
     C(ILAI,ILA5),(ILAI,ILA6),(ILAI,ILA7),(ILAI,ILA8),(ILAI,ILA9),
     C(ILAI,ILA10),(ILAI,ILA11),(ILAI,ILA12),(ILAI,ILA13),(ILAI,ILA14),
     C(ILAI,ILA15),(ILAI,ILA16),(ILAI,ILA17),(ILAI,ILA18),(ILAI,ILA19),
     C(ILAI,ILA20),(ILAI,ILA21),(ILAI,ILA22),(ILAI,ILA23),(
     CILAI,ILA24),(ILAI,ILA25),(ILAI,ILA26),(ILAI,ILA27),
     C(ILAI,ILA28),(ILAI,ILA29)
      COMMON A41,U,DELT,UDOT,V,VDOT,R,RDOT,P,PDOT,Q,QDOT,THETA,T,AM,
     CCDP,LZ,A11,A21,A31,X,AXST,AYST,
     CZ1,CDX,RZ,DD,CDY,WW,A22,FDP,PHI,RDS,FSP,FSS,CDZP,A33,RX,F1,F2,
     CF3,LF,LC,FB,WE,RY,L3,A34,Z,BETA,HYSESX,XST,HYSESY,YST,UMAX,XGX,
     CXDX,LMLL,HEAD,YHIGH,SPCLIN,AICOA,YO,ICOA,SEAST
      COMMON /DATA/IARR
      NAMELIST
      CALL AC(IARR)
 1111 CONTINUE
      CALL RESET(1000)
      CALL DTINIT(1,ITEXT,29,IER)
      CALL DCINIT(1,ISWA,40,IER)
      CALL DTINIT(2,IPLOT,29,IER)
      CALL DGINIT(2,IL,29,IER)
      ENCODE(96,1,TITLEO)
    1 FORMAT(             REAL TIME                 )
      CALL TEXTO(1,TITLEC,24,10,1,3,3,IER)
      IF(IER.NE.0)OUTPUT(IO1) IER,      1
      ENCODE(96,2,TITLE1)
    2 FORMAT(      MOTION ANALYSIS PROGRAM        )
      CALL TEXTO(1,TITLE1,24,13,1,3,3,IER)
      IF(IER.NE.0)OUTPUT(IO1) IER,      2
      I=1000000
      CALL DELAY
      V=0.
      W=0.
      R=0.
      P=0.
```

```
      Q=0.
      X0=0.
      Y0=0.
      ZZ=0.
      PSI=0.
      PHI=0.
      THETA=0.
      UDOT=0.
      VDOT=0.
      WDOT=0.
      RDOT=0.
      PDOT=0.
      QDOT=0.
      TIME=0.
      DO 3131 I=1,50
3131  F(I)=0.
3001  CONTINUE
      SEAST=4
      UMAX=099.
      VMAX=10.
      RMAX=0.125
      PMAX=.04
      QMAX=.02
      UDOTM=1.8
      VDOTM=3.6
      RDOTM=0.040
      PDOTM=.1
      QDOTM=.02
      A33=294116.
      A34=588236.
      SLIP=0.
      LRR=50.
      LR=30.
      LZ=30.
      L3=100.
      LD=3.
      WE=100.
      PB=240.
      LP=5.
      CDP=9.174
      CDZ=1000000.
      CDZP=30000.
      RX=182200000.
      RY=815899900.
      W=087.
      TMAX=360000
      T7MAX=90000
      T8MAX=90000
```

```
      T9MAX=90000
      T10MAX=90000
      AM=170364.
      CDY=22500.0
      A22=1.
      CC=110.8
      WW=22.2
      RZ=95270000.
      AYST=20000.
      AXST=20000.
 3002 CONTINUE
      ZZZZ=0.5263
      DELTA=0.0E4
      SFZZX=-.1
      Z1=2.
      Z2=0.5263
      Z3=-0.5263
      Z4=0.02
      Z5=-0.02
      Z6=-.05
      Z11=3.1416
      Z12=0.2
      XST=0.75
      YST=0.75
      XREPET(1)=XST
      YREPET(1)=YST
      SXREPET=XST
      SYREPET=YST
      V=0.
      R=0.
      W=0.
      XO=0.
      YO=0.
      ZZ=0.
      PHI=0.
      THETA=0.02
      PSI=0.
      UDOT=0.
      VDOT=0.
      RDOT=0.
      WDOT=0.
      PDOT=0.
      QDOT=0.
      SF1=12.
      SF14=0.9
      JJ=0
      JP=0
      JQ=0
```

```
JR=0
JS=0
CTIME=0.
PTIME=0.
QTIME=0.
RTIME=0.
STIME=0.
CYREPET=.3
PYREPET=.1
QYREPET=.3
RYREPET=.1
JD=0
JE=0
JF=0
JG=0
JH=0
DTIME=0.0
ETIME=0.0
FTIME=0.0
GTIME=0.0
HTIME=0.0
DYREPET=0.7
EYREPET=0.5
FYREPET=0.7
GYREPET=0.5
HYREPET=0.9
UMUL=4.0
IPR(1)=230
IPR(2)=110
IPRD=110
IPRE=110
IPRF=110
IPRG=110
IPRH=110
IPRO=110
IPRP=110
IPRQ=110
IPRR=110
IPRS=110
ICCA=0
ICC=2
AICCA=10.
X=-100.
TIME=0.0
P=0.0
Q=0.0
RHO=2.0
G=32.2
```

```
      AXO=25.
      AYO=25.
      TINT=1000.
      XGX=24.
      XDX=24.
      ZAB=-3.1416
      YHIGH=0.1
      IN=0
      J(1)=0
      J(2)=0
      TIM(2)=0.
      YREPET(2)=0.9
      APRINT=5.
      K=2900.0
      KK=3.0
      DFCCM2=0.0
      IF((10.0*IARR(1)/(2**23).GT.-.8).AND.(10.0*IARR(1)/(2**23)
     C.LT..5))GO TO 22
 1112 OUTPUT(101) CHANGES,,*+ C/R
      INPUT(101)
      IF(IER.NE.0)OUTPUT(101) IER,      3
   22 CONTINUE
  650 CALL DTINIT(1,ITEXT,29,IER)
      CALL DGINIT(1,ISWA,40,IER)
      CALL DTINIT(2,IPLOT,29,IER)
      CALL DGINIT(2,IL,29,IER)
      IVIEWB(1)=IHEAD(1,10)
      IVIEWB(2)=IPACK(0.0,1.0,0)
      IVIEWB(3)=IPACK(0.0,0.5,0)
      IVIEWB(4)=IPACK(-1.0,0.0,0)
      IVIEWB(5)=IPACK(-0.5,0.0,1)
      IVIEWB(6)=IPACK(0.5,0.0,0)
      IVIEWB(7)=IPACK(1.0,0.0,1)
      IVIEWB(8)=IPACK(0.75,1.0,0)
      IVIEWB(9)=IPACK(0.75,0.5,1)
      IVIEWB(10)=IPACK(0.5,0.75,0)
      IVIEWB(11)=IPACK(1.0,0.75,1)
      IVIEWB(12)=0
      CALL GRAPHO(1,IVIEWB,12,1,IER)
      IVIEWZ(1)=IHEAD(0,10)
      IVIEWZ(2)=IPACK(-.9,0.05,0)
      IVIEWZ(3)=IPACK(-.9,.99,1)
      IVIEWZ(4)=IPACK(-.9,.9,0)
      IVIEWZ(5)=IPACK(0.0,.9,1)
      IVIEWZ(6)=IPACK(-.9,.7,0)
      IVIEWZ(7)=IPACK(0. ,.7,1)
      IVIEWZ(8)=IPACK(-.9,.5,0)
      IVIEWZ(9)=IPACK(.0,.5,1)
```

```
      IVIEWZ(10)=IPACK(-.9,.3,0)
      IVIEWZ(11)=IPACK(.0,.3,1)
      IVIEWZ(12)=IPACK(-.9,.1,0)
      IVIEWZ(13)=IPACK(0.,.1,1)
      IVIEWZ(14)=IPACK(0.0,0.01,0)
      IVIEWZ(15)=IPACK(0.0,0.01,1)
      IVIEWZ(16)=IPACK(0.0,0.03,1)
      IVIEWZ(17)=0
      CALL GRAPHC(1,IVIEWZ,17,2,IER)
  501 FORMAT( PRESENT ,2F10.2,          ,F4.2,      ,F8.2,            ,F6.3,
     C         ,F6.3)
  525 FORMAT( MAXIMUM ,2F10.2,          ,F4.2,      ,F8.2,            ,F6.3,
     C         ,F6.3)
  526 FORMAT( MINIMUM ,2F10.2,          ,F4.2,      ,F8.2,            ,F6.3,
     C         ,F6.3)
  507 FORMAT( PRESENT ,F8.2,F10.3,           ,F4.2,   ,F8.3,                 ,
     CF5.3,            ,F5.3)
  527 FORMAT( MAXIMUM ,F8.2,F10.3,           ,F4.2,   ,F8.3,                 ,
     CF5.3,            ,F5.3)
  528 FORMAT( MINIMUM ,F8.2,F10.3,           ,F4.2,   ,F8.3,                 ,
     CF5.3,            ,F5.3)
  511 FORMAT( PRESENT ,F7.3,F10.3,       ,F10.2,      ,F8.3,           ,F6.3,
     C       ,F6.3)
  529 FORMAT( MAXIMUM ,F7.3,F10.3,       ,F10.2,      ,F8.3,           ,F6.3,
     C       ,F6.3)
  530 FORMAT( MINIMUM ,F7.3,F10.3,       ,F10.2,      ,F8.3,           ,F6.3,
     C       ,F6.3)
  516 FORMAT(F8.0,   ,F4.2,          ,F4.3,      ,F6.2,           ,F5.3,
     C   ,F7.2, / ,F5.2,     ,F5.2,    ,F5.2,      ,F5.2)
      ENCODE(96,500,ILA1)
  500 FORMAT(                                     POSITION )
      CALL TEXTC(2,ILA1,24,7,4,1,3,IER)
      ENCODE(96,502,ILA2)
  502 FORMAT(               X          Y          ZZ          PSI
     CPHI          THETA )
      CALL TEXTC(2,ILA2,24,8,4,1,3,IER)
      ENCODE(96,505,ILA3)
  505 FORMAT(                                     VELOCITY )
      CALL TEXTC(2,ILA3,24,15,4,1,3,IER)
      ENCODE(96,517,ILA4)
  517 FORMAT(               L          V          W           R
     CP             Q )
      CALL TEXTC(2,ILA4,24,16,4,1,3,IER)
      ENCODE(96,509,ILA5)
  509 FORMAT(                                   ACCELERATION )
      CALL TEXTC(2,ILA5,24,23,4,1,3,IER)
      ENCODE(96,512,ILA6)
  512 FORMAT(            UDOT        VDOT         WDOT        RDOT        PD
```

```
    CCT      CCCT )
     CALL TEXTC(2,ILA6,24,24,4,1,3,IER)
     ENCODE(96,514,ILA7)
 514 FORMAT(  CONTROL INPUTS          TIME                          NAV
    CIGATION DATA )
     CALL TEXTO(1,ILA7,24,32,1,1,3,IER)
     ENCODE(96,515,ILA8)
 515 FORMAT( THRUST   RUDDER      DELT   ELAPSED      PITCH   HEADING/R
    CATE   DRIFT   ROLL   SPEED )
     CALL TEXTO(1,ILA8,24,33,1,1,3,IER)
     IF(IER.NE.0)OUTPUT(101) IER,      12
     ENCODE(96,532,ILA9)
 532 FORMAT( THRUST1      THRUST2     THRUST3     THRUST4  SEA STATE
    CCASST     KK        SPCCM )
     CALL TEXTC(1,ILA9,24,36,1,1,3,IER)
     IF(IER.NE.0)OUTPUT(101) IER,      13
     ENCODE(96,2050,ILA10)
2050 FORMAT(      VARIABLE DYNAMICS
    CSEA TRACK )
     CALL TEXTO(1,ILA10,24,3,15,1,3,IER)
     ENCODE(96,2051,ILA11)
2051 FORMAT(  L  (HEAVY) )
     CALL TEXTO(1,ILA11,24,5,1,1,3,IER)
     ENCODE(96,2054,ILA14)
2054 FORMAT( LDOT(LIGHT) )
     CALL TEXTO(1,ILA14,24,6,1,1,3,IER)
     ENCODE(96,2052,ILA12)
2052 FORMAT(  V  (HEAVY) )
     CALL TEXTO(1,ILA12,24,9,1,1,3,IER)
     ENCODE(96,2055,ILA15)
2055 FORMAT( VDOT(LIGHT) )
     CALL TEXTC(1,ILA15,24,10,1,1,3,IER)
     ENCODE(96,2053,ILA13)
2053 FORMAT(  R  (HEAVY) )
     CALL TEXTO(1,ILA13,24,12,1,1,3,IER)
     ENCODE(96,2056,ILA16)
2056 FORMAT( RDOT(LIGHT) )
     CALL TEXTC(1,ILA16,24,13,1,1,3,IER)
     ENCODE(96,2057,ILA17)
2057 FORMAT(                                        ZERO PITCH )
     CALL TEXTO(1,ILA17,24,21,45,1,3,IER)
     ENCODE(96,2058,ILA18)
2058 FORMAT( DATUM LINE )
     CALL TEXTC(1,ILA18,24,22,84,1,3,IER)
     ENCODE(96,2059,ILA19)
2059 FORMAT(  VARIABLE STATUS )
     CALL TEXTO(2,ILA19,24,4,28,2,3,IER)
     IF(IER.NE.0)OUTPUT(101) IER,      6
```

```
      ENCODE(96,3010,ILA21)
 3010 FORMAT(   F   (HEAVY) )
      CALL TEXTO(1,ILA21,24,16,1,1,3,IER)
      ENCODE(96,3011,ILA22)
 3011 FORMAT( FDCT(LIGHT) )
      CALL TEXTO(1,ILA22,24,17,1,1,3,IER)
      ENCODE(96,3012,ILA23)
 3012 FORMAT(   C   (HEAVY) )
      CALL TEXTO(1,ILA23,24,19,1,1,3,IER)
      ENCODE(96,3013,ILA24)
 3013 FORMAT( CDCT(LIGHT) )
      CALL TEXTO(1,ILA24,24,20,1,1,3,IER)
      ENCODE(96,3,ILA25)
    3 FORMAT( WATERLINE )
      CALL TEXTO(1,ILA25,24,24,1,1,3,IER)
  531 FORMAT(F8.1,        ,F8.1,        ,F8.1,          ,F8.1,          ,F4.2,
     C ,F4.2,        ,F8.1,             ,F5.2)
      CALL WRITECLOCK(0)
      CALL STARTCLOCK
      A41=10.0*IARR(2)/2**23
      CALL COMPUTE
  101 CONTINUE
  149 ICO=ICO+1
  621 CONTINUE
      IF(SENSE SWITCH 6)9999,9998
 9999 Z=ZZZZ
      ZI=Z
      GO TO 792
 9998 CONTINUE
      Z=1.4*A41
      ZI=Z
      GO TO 744
  744 CONTINUE
      IF(Z.GT.Z2)GO TO 790
      IF(Z.LT.Z3)GO TO 791
      IF(Z.GT.Z4)GO TO 792
      IF(Z.LT.Z5)GO TO 792
      Z=0.0
      ZI=Z
      GO TO 792
  790 Z=Z2
      ZI=Z
      GO TO 792
  791 Z=Z3
      ZI=Z
      GO TO 792
  792 CONTINUE
      IF(U.LT.70.)ZBRO=Z2
```

```
      IF((U.GE.70.).AND.(U.LT.87.))ZBRO=.3491
      IF((U.GE.87.).AND.(U.LT.104.))GO TO 4005
      IF((U.GE.104.).AND.(U.LT.121.))GO TO 4006
      IF((U.GE.121.).AND.(U.LT.138.))GO TO 4007
      IF(U.GE.138.)ZBRO=.0349
      GO TO 4008
4005  ZBRO=.3491*17/(U-70)
      GO TO 4008
4006  ZBRO=.1745*17/(U-87)
      GO TO 4008
4007  ZBRO=.0873*17/(U-104)
4008  CONTINUE
      IF(ABS(Z).LT.ZBRO)GO TO 4009
      ENCODE(96,4010,ILA9)
4010  FORMAT(       B  R  E  A  C  H                )
      CALL TEXTO(1,ILA9,24,37,1,1,3,IER)
4009  CONTINUE
 622  CONTINUE
      T7=T7MAX*(.5+(5.0*IARR(3)/2**23))
      T8=T8MAX*(.5+(5.0*IARR(4)/2**23))
      T9=T9MAX*(.5+(5.0*IARR(5)/2**23))
      T10=T9
      TM=T7+T8+T9+T10
      IF(SENSE SWITCH 3)4000,4001
4001  CASST=0
      FCOM=TM
      GO TO 4002
4000  CASST=1
      UCOM=200*(.5+(5.0*IARR(6)/2**23))
      DUCOM = UCOM-U
      DFCOM1=K*DUCOM
      DFCOM2=DFCOM2+DUCOM*DELT
      DFCOM=DFCOM1+KK*DFCOM2
      FCOM=DFCOM+TM
      IF(FCOM.GT.TMAX)GO TO 4003
      IF(FCOM.LT.0.0)GO TO 4004
      GO TO 4002
4003  FCOM=TMAX
      GO TO 4002
4004  FCOM=0.0
      GO TO 4002
4002  TDOT=FCOM-T
      T=T+TDOT*DELT
      IF(T.GT.TMAX)T=TMAX
      IF(T.LT.0.0)T=0.0
      X1=PSI/6.283185
      IX=X1
      AIX=X1-IX
```

```
      IF(AIX.LT.0.0)GO TO 10
      HEAD=AIX*360.
      GO TO 11
 10   AIX1=1.+AIX
      HEAD=AIX1*360.
 11   CONTINUE
      ARATE=(360.*R)/6.283185
      AROLL=(360.*PHI)/6.283185
      APITCH=(360.*THETA)/6.283185
      SPEED=.5741*U
      SPCOM=.5741*UCOM
      IF(SENSE SWITCH 2)176,175
175   CALL CNEMCT
176   CONTINUE
177   CONTINUE
      BETA=SLIP*R
      VS=((U*U)+(V*V))**.5
      PSI=PSI+DELT*R
      XODOT=U*COS(PSI)-V*SIN(PSI)
      YODOT=U*SIN(PSI)+V*COS(PSI)
      XO=XO+DELT*XODOT
      YO=YO+DELT*YODOT
      SESX=XST+(YO/AXST)
      SESY=YST+(XO/AYST)
      IF(H(1).GT.XO)GO TO 700
      H(1)=XO
700   IF(H(2).GT.YO)GO TO 701
      H(2)=YO
701   IF(H(3).GT.ZZ)GO TO 702
      H(3)=ZZ
702   IF(H(4).GT.PSI)GO TO 703
      H(4)=PSI
703   IF(H(5).GT.PHI)GO TO 704
      H(5)=PHI
704   IF(H(6).GT.THETA)GO TO 705
      H(6)=THETA
705   IF(H(7).GT.U)GO TO 706
      H(7)=U
706   IF(H(8).GT.V)GO TO 707
      H(8)=V
707   IF(H(9).GT.W)GO TO 708
      H(9)=W
708   IF(H(10).GT.R)GO TO 709
      H(10)=R
709   IF(H(11).GT.P)GO TO 710
      H(11)=P
710   IF(H(12).GT.Q)GO TO 711
      H(12)=Q
```

```
711 IF(H(13).GT.UDOT)GO TO 712
    H(13)=UDOT
712 IF(H(14).GT.VDOT)GO TO 713
    H(14)=VDOT
713 IF(H(15).GT.WDOT)GO TO 714
    H(15)=WDOT
714 IF(H(16).GT.RDOT)GO TO 715
    H(16)=RDOT
715 IF(H(17).GT.PDOT)GO TO 716
    H(17)=PDOT
716 IF(H(18).GT.QDOT)GO TO 718
    H(18)=QDOT
718 IF(H(30).LT.XO)GO TO 719
    H(30)=XO
719 IF(H(31).LT.YO)GO TO 720
    H(31)=YO
720 IF(H(32).LT.ZZ)GO TO 721
    H(32)=ZZ
721 IF(H(33).LT.PSI)GO TO 722
    H(33)=PSI
722 IF(H(34).LT.PHI)GO TO 723
    H(34)=PHI
723 IF(H(35).LT.THETA)GO TO 724
    H(35)=THETA
724 IF(H(36).LT.U)GO TO 725
    H(36)=U
725 IF(H(37).LT.V)GO TO 726
    H(37)=V
726 IF(H(38).LT.W)GO TO 727
    H(38)=W
727 IF(H(39).LT.R)GO TO 728
    H(39)=R
728 IF(H(40).LT.P)GO TO 729
    H(40)=P
729 IF(H(41).LT.Q)GO TO 730
    H(41)=Q
730 IF(H(42).LT.UDOT)GO TO 731
    H(42)=UDOT
731 IF(H(43).LT.VDOT)GO TO 732
    H(43)=VDOT
732 IF(H(44).LT.WDOT)GO TO 733
    H(44)=WDOT
733 IF(H(45).LT.RDOT)GO TO 734
    H(45)=RDOT
734 IF(H(46).LT.PDOT)GO TO 735
    H(46)=PDOT
735 IF(H(47).LT.QDOT)GO TO 736
    H(47)=QDOT
```

```
 736 CONTINUE
     IF(SENSE SWITCH 1)102,105
 102 CONTINUE
     IIX=IIX+1
     IF(IIX.LT.APRINT)GO TO 7654
     IIX=0
     WRITE(6,100)TIME,U,V,R,PHI,THETA,XO,YO
 100 FORMAT(6F12.5,2F8.1)
7654 CONTINUE
105  CONTINUE
     IF(ICC.NE.3)GO TO 623
 602 ENCODE(96,516,ILAP)T,Z,DELT,TIME,APITCH,HEAD,ARATE,BETA,ARCLL,SPEE
    CD
     CALL TEXTC(1,ILAP,24,34,1,1,3,IER)
 623 CONTINUE
 58  CONTINUE
     CALL BCAT
2071 IF((10.0*IARR(1)/(2**23)).GT.-.8)GO TO 2080
     GO TO 2072
2080 IF((10.0*IARR(1)/(2**23)).GT..5)GO TO 1111
     IF ((10.0*IARR(1)/(2**23)).LT.-.4)GO TO 23
     GO TO 1111
 23  CALL WRITECLOCK(0)
     GO TO 2071
2072 CONTINUE
     IF(ICC.NE.15)GO TO 624
     IIPR(1)=IPR(1)-1
     J(1)=J(1)+1
     IJ(1)=4+J(1)
     IJA(1)=IJ(1)+1
     IVIEWAA(1)=IHEAD(0,10)
     IVIEWAA(2)=IPACK(.0,.0,0)
     IVIEWAA(3)=IPACK(XREPET(1),YREPET(1),0)
     IVIEWAA(4)=IPACK(XREPET(1),YREPET(1),1)
     IVIEWAA(IJ(1))=IPACK(SESX,SESY,1)
     DO 2015 I=IJA(1),IPR(1)
2015 IVIEWAA(I)=0
     IF(IJA(1).LT.IIPR(1))GO TO 2017
2018 CONTINUE
     DO 2016 I=1,IPR(1)
2016 IVIEWAA(I)=0
     XREPET(1)=SESX
     YREPET(1)=SESY
     J(1)=0
     IJ(1)=0
     IJA(1)=0
2017 CONTINUE
     CALL GRAFFC(1,IVIEWAA,IPR(1),15,IER)
```

```
  624 CONTINUE
  606 CONTINUE
      IF(ICC.NE.10)GO TO 630
      IIPR(2)=IPR(2)-1
      J(2)=J(2)+1
      IJ(2)=4+J(2)
      IJA(2)=IJ(2)+1
      TIM(2)=TIM(2)+DELT
      TIMPLT(2)=(TIM(2)/SF1)-SF14
      XPLOT(2)=TIMPLT(2)
      YPLOT(2)=(U/(10.*UMAX))+.9
      IVIEWC(1)=IHEAD(0,10)
      IVIEWC(2)=IPACK(.0,.0,0)
      IVIEWC(3)=IPACK(-.9,YREPET(2),0)
      IVIEWC(4)=IPACK(-.9,YREPET(2),1)
      IVIEWC(IJ(2))=IPACK(XPLOT(2),YPLOT(2),1)
      DO 2020 I=IJA(2),IPR(2)
 2020 IVIEWC(I)=0
      IF(IJA(2).LT.IIPR(2))GO TO 2032
 2038 CONTINUE
      DO 2044 I=1,IPR(2)
 2044 IVIEWC(I)=0
      TIM(2)=0.
      YREPET(2)=YPLOT(2)
      J(2)=0
      IJ(2)=0
      IJA(2)=0
 2032 CONTINUE
      CALL GRAFFC(1,IVIEWC,IPR(2),10,IER)
  630 CONTINUE
      IF(ICC.NE.11)GO TO 631
      IIPRD=IPRD-1
      JD=JD+1
      IJD=4+JD
      IJAD=IJD+1
      DTIME=DTIME+DELT
      DTIMPLT=(DTIME/SF1)-SF14
      DXPLOT=DTIMPLT
      DYPLOT=(V/(10.*VMAX))+0.7
      IVIEWD(1)=IHEAD(0,10)
      IVIEWD(2)=IPACK(.0,.0,0)
      IVIEWD(3)=IPACK(-.9,DYREPET,0)
      IVIEWD(4)=IPACK(-.9,DYREPET,1)
      IVIEWD(IJD)=IPACK(DXPLOT,DYPLOT,1)
      DO 2021 I=IJAD,IPRD
 2021 IVIEWD(I)=0
      IF(IJAD.LT.IIPRD)GO TO 2033
 2039 CONTINUE
```

```
      DO 2045 I=1,IPRD
 2045 IVIEWD(I)=0
      DTIME=0.
      DYREPET=DYPLOT
      JD=0
      IJD=0
      IJAD=0
 2033 CONTINUE
      CALL GRAPHO(1,IVIEWD,IPRD,11,IER)
  631 CONTINUE
      IF(ICC.NE.12)GO TO 632
      IIPRE=IPRE-1
      JE=JE+1
      IJE=4+JE
      IJAE=IJE+1
      ETIME=ETIME+DELT
      ETIMPLT=(ETIME/SF1)-SF14
      EXPLOT=ETIMPLT
      EYPLOT=(F/(10.*RMAX))+0.5
      IVIEWE(1)=IHEAD(0,10)
      IVIEWE(2)=IPACK(.0,.0,0)
      IVIEWE(3)=IPACK(-.9,EYREPET,0)
      IVIEWE(4)=IPACK(-.9,EYREPET,1)
      IVIEWE(IJE)=IPACK(EXPLOT,EYPLOT,1)
      DO 2022 I=IJAE,IPRE
 2022 IVIEWE(I)=0
      IF(IJAE.LT.IIPRE)GO TO 2034
 2040 CONTINUE
      DO 2046 I=1,IPRE
 2046 IVIEWE(I)=0
      ETIME=0.
      EYREPET=EYPLOT
      JE=0
      IJE=0
      IJAE=0
 2034 CONTINUE
      CALL GRAPHO(1,IVIEWE,IPRE,12,IER)
  632 CONTINUE
      IF(ICC.NE.6)GO TO 626
      IIPRF=IPRF-1
      JF=JF+1
      IJF=4+JF
      IJAF=IJF+1
      FTIME=FTIME+DELT
      FTIMPLT=(FTIME/SF1)-SF14
      FXPLOT=FTIMPLT
      FYPLOT=(VDOT/(10.*VDOTM))+0.7
      IVIEWF(1)=IHEAD(0,3)
```

```
      IVIEWF(2)=IPACK(.0,.0,0)
      IVIEWF(3)=IPACK(-.9,FYREPET,0)
      IVIEWF(4)=IPACK(-.9,FYREPET,1)
      IVIEWF(IJF)=IPACK(FXFLOT,FYPLOT,1)
      DO 2023 I=IJAF,IPRF
 2023 IVIEWF(I)=0
      IF(IJAF.LT.IIPRF)GO TO 2035
 2041 CONTINUE
      DO 2047 I=1,IPRF
 2047 IVIEWF(I)=0
      FTIME=0.
      FYREPET=FYPLOT
      JF=0
      IJF=0
      IJAF=0
 2035 CONTINUE
      CALL GRAPHO(1,IVIEWF,IPRF,6,IER)
  626 CONTINUE
      IF(ICO.NE.7)GO TO 627
      IIPRG=IPRG-1
      JG=JG+1
      IJG=4+JG
      IJAG=IJG+1
      GTIME=GTIME+DELT
      GTIMPLT=(GTIME/SF1)-SF14
      GXPLOT=GTIMPLT
      GYPLOT=(RCCT/(10.*RCCTM))+0.5
      IVIEWG(1)=IHEAD(0,3)
      IVIEWG(2)=IPACK(.0,.0,0)
      IVIEWG(3)=IPACK(-.9,GYREPET,0)
      IVIEWG(4)=IPACK(-.9,GYREPET,1)
      IVIEWG(IJG)=IPACK(GXPLOT,GYPLOT,1)
      DO 2024 I=IJAG,IPRG
 2024 IVIEWG(I)=0
      IF(IJAG.LT.IIPRG)GO TO 2036
 2042 CONTINUE
      DO 2048 I=1,IPRG
 2048 IVIEWG(I)=0
      GTIME=0.
      GYREPET=GYPLOT
      JG=0
      IJG=0
      IJAG=0
 2036 CONTINUE
      CALL GRAPHO(1,IVIEWG,IPRG,7,IER)
  627 CONTINUE
      IF(ICO.NE.5)GO TO 625
      IIPRF=IPRF-1
```

```
      JH=JH+1
      IJH=4+JH
      IJAH=IJH+1
      HTIME=HTIME+DELT
      HTIMPLT=(HTIME/SF1)-SF14
      HXPLOT=HTIMPLT
      HYPLOT=(UDCT/(10.*UDCTM))+0.9
      IVIEWH(1)=IHEAD(0,3)
      IVIEWH(2)=IPACK(.0,.0,0)
      IVIEWH(3)=IPACK(-.9,HYREPET,0)
      IVIEWH(4)=IPACK(-.9,HYREPET,1)
      IVIEWH(IJH)=IPACK(HXPLOT,HYPLOT,1)
      DO 2025 I=IJAH,IPRH
2025  IVIEWH(I)=0
      IF(IJAH.LT.IIPRH)GO TO 2037
2043  CONTINUE
      DO 2049 I=1,IPRH
2049  IVIEWH(I)=0
      HTIME=0.
      HYREPET=HYPLOT
      JH=0
      IJH=0
      IJAH=0
2037  CONTINUE
      CALL GRAFFO(1,IVIEWH,IPRH,5,IER)
 625  CONTINUE
      IF(ICO.NE.8)GO TO 628
      IIPRO=IPRO-1
      JO=JO+1
      IJO=4+JO
      IJAO=IJO+1
      OTIME=OTIME+DELT
      OTIMPLT=(OTIME/SF1)-SF14
      OXPLOT=OTIMPLT
      OYPLOT=(PDCT/(10.*PDCTM))+.3
      IVIEWO(1)=IHEAD(0,3)
      IVIEWO(2)=IPACK(.0,.0,0)
      IVIEWO(3)=IPACK(-.9,OYREPET,0)
      IVIEWO(4)=IPACK(-.9,OYREPET,1)
      IVIEWO(IJO)=IPACK(OXPLOT,OYPLOT,1)
      DO 2060 I=IJAO,IPRO
2060  IVIEWO(I)=0
      IF(IJAO.LT.IIPRO)GO TO 2065
      DO 2090 I=1,IPRO
2090  IVIEWO(I)=0
      OTIME=0.
      OYREPET=OYPLOT
      JO=0
```

```
      IJC=C
      IJAC=C
 2065 CONTINUE
      CALL GRAPHC(1,IVIEWC,IPRC,8,IER)
  628 CONTINUE
      IF(ICC.NE.9)GO TO 629
      IIPRP=IPRP-1
      JP=JP+1
      IJP=4+JP
      IJAP=IJP+1
      PTIME=PTIME+DELT
      PTIMPLT=(PTIME/SF1)-SF14
      PXPLOT=PTIMPLT
      PYPLOT=(QDOT/(10.*QDOTM))+.1
      IVIEWP(1)=IHEAD(0,3)
      IVIEWP(2)=IPACK(.0,.C,0)
      IVIEWP(3)=IPACK(-.9,PYREPET,0)
      IVIEWP(4)=IPACK(-.9,PYREPET,1)
      IVIEWP(IJP)=IPACK(PXPLOT,PYPLOT,1)
      DO 2061 I=IJAP,IPRP
 2061 IVIEWP(I)=0
      IF(IJAP.LT.IIPRP)GO TO 2066
      DO 2091 I=1,IPRP
 2091 IVIEWP(I)=0
      PTIME=0.
      PYREPET=PYPLOT
      JP=0
      IJP=C
      IJAP=0
 2066 CONTINUE
      CALL GRAPHC(1,IVIEWP,IPRP,9,IER)
      IF(IER.NE.0)OUTPUT(1C1) IER,       10
  629 CONTINUE
      IF(ICC.NE.13)GO TO 633
      IIPRQ=IPRQ-1
      JQ=JQ+1
      IJQ=4+JQ
      IJAQ=IJQ+1
      QTIME=QTIME+DELT
      QTIMPLT=(QTIME/SF1)-SF14
      QXPLOT=QTIMPLT
      QYPLOT=(F/(10.*PMAX))+.3
      IVIEWQ(1)=IHEAD(0,1C)
      IVIEWQ(2)=IPACK(.C,.C,C)
      IVIEWQ(3)=IPACK(-.9,QYREPET,0)
      IVIEWQ(4)=IPACK(-.9,QYREPET,1)
      IVIEWQ(IJQ)=IPACK(QXPLOT,QYPLOT,1)
      DO 2062 I=IJAQ,IPRQ
```

```
2062 IVIEWQ(I)=C
     IF(IJAQ.LT.IIPRQ)GO TO 2C67
     DO 2092 I=1,IPRQ
2092 IVIEWQ(I)=C
     QTIME=C.
     CYREPET=CYPLOT
     JQ=0
     IJQ=C
     IJAQ=0
2067 CONTINUE
     CALL GRAPHC(1,IVIEWQ,IPRQ,13,IER)
     IF(IER.NE.C)OUTPUT(ICI) IER,      9-
 633 CONTINUE
     IF(ICC.NE.14)GO TO 634
     IIPRR=IPRR-1
     JR=JR+1
     IJR=4+JR
     IJAR=IJR+1
     RTIME=RTIME+DELT
     RTIMPLT=(RTIME/SF1)-SF14
     RXPLOT=RTIMPLT
     RYPLOT=(C/(10.*QMAX))+.1
     IVIEWR(1)=IHEAD(0,1C)
     IVIEWR(2)=IPACK(.C,.C,C)
     IVIEWR(3)=IPACK(-.9,RYREPET,0)
     IVIEWR(4)=IPACK(-.9,RYREPET,1)
     IVIEWR(IJR)=IPACK(RXPLOT,RYPLOT,1)
     DO 2063 I=IJAR,IPRR
2063 IVIEWR(I)=0
     IF(IJAR.LT.IIPRR)GO TO 2C68
     DO 2C73 I=1,IPRR
2073 IVIEWR(I)=C
     RTIME=C.
     RYREPET=RYPLOT
     JR=0
     IJR=C
     IJAR=C
2068 CONTINUE
     CALL GRAPHC(1,IVIEWR,IPRR,14,IER)
     IF(IER.NE.C)OUTPUT(ICI) IER,      8
 634 CONTINUE
     IF(ICC.NE.4)GO TO 635
     IIPRS=IPRS-1
     JS=JS+1
     IJS=4+JS
     IJAS=IJS+1
     STIME=STIME+DELT
     STIMPLT=(STIME/SF1)-SF14
```

```
      SXPLOT=FYSESX
      SYPLOT=FYSESY
      IVIEWS(1)=IHEAD(0,1)
      IVIEWS(2)=IPACK(.0,.0,0)
      IVIEWS(3)=IPACK(SXREPET,SYREPET,0)
      IVIEWS(4)=IPACK(SXREPET,SYREPET,1)
      IVIEWS(IJS)=IPACK(SXPLOT,SYPLOT,1)
      DO 2064 I=IJAS,IPRS
 2064 IVIEWS(I)=0
      IF(IJAS.LT.IIPRS)GO TO 2069
      DO 2074 I=1,IPRS
 2074 IVIEWS(I)=0
      STIME=0.
      SXREPET=FYSESX
      SYREPET=FYSESY
      JS=0
      IJS=0
      IJAS=0
 2069 CONTINUE
      CALL GRAPHC(1,IVIEWS,IPRS,4,IER)
      IF(IER.NE.0)OUTPUT(101) IER,      5
  635 CONTINUE
      IF(ICC.NE.16)GO TO 636
  653 ENCODE(96,501,ILAP)YO,XO,ZZ,PSI,PHI,THETA
  600 CALL TEXTO(2,ILAP,24,10,4,1,3,IER)
  636 CONTINUE
      IF(ICC.NE.17)GO TO 637
  110 ENCODE(96,526,ILAP)H(31),H(30),H(32),H(33),H(34),H(35)
      CALL TEXTO(2,ILAP,24,13,4,1,3,IER)
  637 CONTINUE
      IF(ICC.NE.18)GO TO 638
  108 ENCODE(96,525,ILAP)H(2),H(1),H(3),H(4),H(5),H(6)
      CALL TEXTO(2,ILAP,24,12,4,1,3,IER)
  638 CONTINUE
      IF(ICC.NE.19)GO TO 639
  601 ENCODE(96,507,ILAP)U,V,W,R,P,Q
      CALL TEXTO(2,ILAP,24,18,4,1,3,IER)
  639 CONTINUE
      IF(ICC.NE.20)GO TO 640
  115 ENCODE(96,528,ILAP)H(36),H(37),H(38),H(39),H(40),H(41)
      CALL TEXTO(2,ILAP,24,21,4,1,3,IER)
  640 CONTINUE
      IF(ICC.NE.21)GO TO 641
  113 ENCODE(96,527,ILAP)H(7),H(8),H(9),H(10),H(11),H(12)
      CALL TEXTO(2,ILAP,24,20,4,1,3,IER)
  641 CONTINUE
      IF(ICC.NE.22)GO TO 642
  604 ENCODE(96,511,ILAP)UDOT,VDOT,WDOT,RDOT,PDOT,QDOT
```

```
      CALL TEXTC(2,ILAP,24,26,4,1,3,IER)
  642 CONTINUE
      IF(ICC.NE.23)GO TO 643
  120 ENCODE(96,530,ILAP)H(42),H(43),H(44),H(45),H(46),H(47)
      CALL TEXTC(2,ILAP,24,29,4,1,3,IER)
  643 CONTINUE
      IF(ICC.NE.24)GO TO 644
  118 ENCODE(96,529,ILAP)H(13),H(14),H(15),H(16),H(17),H(18)
      CALL TEXTC(2,ILAP,24,28,4,1,3,IER)
  644 CONTINUE
      IF(ICC.NE.25)GO TO 645
      ENCODE(96,531,ILAP)T7,T8,T9,T10,SEAST,CASST,KK,SPCCM
      CALL TEXTC(1,ILAP,24,37,1,1,3,IER)
  645 CONTINUE
      CALL READCLOCK(N)
      CALL WRITECLOCK(0)
      IF(SENSE SWITCH 5)740,741
  740 DELT=DELTA
      GO TO 742
  741 CONTINUE
      DELT=N/TINT
  742 CONTINUE
      TIME=TIME+DELT
  610 IF(ICC.LE.24)GO TO 101
      ICC=0
      GO TO 101
      END
      SUBROUTINE BOAT
      DIMENSION IVIEWA(75)
      COMMON A41,U,DELT,UDOT,V,VDOT,R,RDOT,P,PDOT,Q,QDOT,THETA,T,AM,
     CCDP,LZ,A11,A21,A31,X,AXST,AYST,
     CZI,CDX,RZ,OO,CDY,WW,A22,RDP,PHI,RDS,FSP,FSS,CDZP,A33,RX,F1,F2,
     CF3,LP,LC,PB,WE,RY,L3,A34,Z,BETA,HYSESX,XST,HYSESY,YST,UMAX,XGX,
     CXDX,LMLL,HEAD,YHIGH,SPOLIN,AICOA,YO,ICOA,SEAST
      ABSYO=ABS(YO)
      HOR3=X1XB-.13
      HOR5=X1XA+.13
      YAWPX=2.5*R
      YAWPY=THETA
      X1X=.45*COS(PHI)
      Y1Y=.45*SIN(PHI)
      X1XA=(2.5*R)+X1X
      X11=2.5*R
      Y1YA=THETA-Y1Y
      X1XB=(2.5*R)-X1X
      Y1YB=THETA+Y1Y
      X1XC=(2.5*R)+(0.1*SIN(PHI))
      Y1YC=THETA+(0.1*COS(PHI))
```

```
      ICOA=ICCA+1.+(AICOA*(U/UMAX))
      Y1YD=(ICCA/514.)*(-1.0)+(THETA)-0.14
      Y1YE=(ICCA/514.)*(-1.0)+(THETA)-0.14
      X1XD=(X1XB-0.13)-((ICOA/514.)*0.2)
      X1XDL=(X1XE-0.13)-((ICOA/514.)*0.2)-(((ICOA/514.)*.2)*XGX*R)
      X1XE=X1XB-0.13
      X1XEL=X1XB-0.13-(((ICOA/514.)*.2)*XGX*R)
      X1XF=X1XA+0.13
      X1XFR=X1XA+0.13-(((ICCA/514.)*.2)*XDX*R)
      X1XG=(X1XA+0.13)+((ICOA/514.)*0.2)
      X1XGR=(X1XA+0.13)+((ICCA/514.)*0.2)-(((ICOA/514.)*.2)*XDX*R)
      Y1YR=Y1YA-0.14
      Y1YL=Y1YE-0.14
      Y1YFE=Y1YE-(((ICOA/514.)*.2)*((U/UMAX)*(UMUL)))
      REARA1=-0.5
      REARA2=-0.2+Y1Y
      REARB1=-0.5
      REARB2=-0.54+Y1Y
      REARC1=0.5
      REARC2=-0.2-Y1Y
      REARD1=0.5
      REARD2=-0.54-Y1Y
      SPCLIN=((-.24)*(U/UMAX))-.14
      YOREF1=(YC/10000.)*(-1.)
      YOREF=(YC/1000.)*(-1.)
      ROAD4X=RODHED4+(1.5*SIN(YOREF))
      IF(ROAD4X.GT.1.)GO TO 30
      IF(ROAD4X.LT.-1.)GO TO 31
      GO TO 32
   30 CONTINUE
      ROAD4X=1.
      GO TO 32
   31 CONTINUE
      ROAD4X=-1.
      GO TO 32
   32 CONTINUE
      ROAD1X=-.3+RODHED4+(1.5*SIN(YOREF))
      IF(ROAD1X.GT.1.)GO TO 33
      IF(ROAD1X.LT.-1.)GO TO 34
      GO TO 35
   33 CONTINUE
      ROAD1X=1.
      GO TO 35
   34 CONTINUE
      ROAD1X=-1.
      GO TO 35
   35 CONTINUE
      ROAD2X=+.3+RODHED4+(1.5*SIN(YOREF))
```

```
      IF(ROAD2X.GT.1.)GO TO 36
      IF(ROAD2X.LT.-1.)GO TO 37
      GO TO 38
   36 CONTINUE
      ROAD2X=1.
      GO TO 38
   37 ROAD2X=-1.
      GO TO 38
   38 CONTINUE
      RODHED1=180.
      RODHED2=HEAD+180.
      IF(RODHED2.LT.360.)GO TO 42
      RODHED2=RODHED2-360.
42    CONTINUE
      RODDIF=((RODHED1-RODHED2)*3.14)/180.
      RODHED4=SIN(RODDIF)
      ROAD3X=RODHED4+ROAD4X
      IF(ROAD3X.GT.1.)GO TO 39
      IF(ROAD3X.LT.-1.)GO TO 40
      GO TO 41
   39 CONTINUE
      ROAD3X=1.
      GO TO 41
   40 CONTINUE
      ROAD3X=-1.
      GO TO 41
   41 CONTINUE
      YROAD=ABS(ROAD4X)
      ROAD3Y=YHIGH-((YROAD)*YHIGH)
      ROAD1Y=0.01
      ROAD2Y=0.01
      ROAD4Y=0.01
      IF(ABSYC.GT.1500.)GO TO 47
      IF(RODHED2.LT.270.)GO TO 46
      GO TO 47
46    IF(RODHED2.GT.090.)GO TO 45
      GO TO 47
47    CONTINUE
      ROAD1X=-1.
      ROAD2X=-1.
      ROAD3X=-1.
      ROAD4X=-1.
      ROAD1Y=0.
      ROAD2Y=0.
      ROAD3Y=0.
      ROAD4Y=0.
   45 CONTINUE
65    CONTINUE
```

```
182   CONTINUE
      IF(Y1YE.GT.SPDLIN)GO TO 79
      ICCA=0
79    CONTINUE
59    CONTINUE
      IVIEWA(1)=IHEAD(0,10)
      IVIEWA(2)=IPACK(-.5,REARA2,0)
      IVIEWA(3)=IPACK(X1XB,Y1YB,1)
      IVIEWA(4)=IPACK(X1XA,Y1YA,1)
      IVIEWA(5)=IPACK(+.5,REARC2,1)
      IVIEWA(6)=IPACK(-.5,REARA2,1)
      IVIEWA(7)=IPACK(-.5,REARE2,1)
      IVIEWA(8)=IPACK(+.5,REARC2,1)
      IVIEWA(9)=IPACK(+.5,REARC2,1)
      IF(R.GT..02)GO TO 50
      IF(R.LT.-.02)GO TO 51
      IVIEWA(10)=IPACK(X1XC,Y1YEE,0)
      IVIEWA(11)=IPACK(X1XE,Y1YE,1)
      IVIEWA(12)=IPACK(X1XF,Y1YE,0)
      IVIEWA(13)=IPACK(X1XC,Y1YEE,1)
      IVIEWA(14)=IPACK(-1.15,-.14,0)
      IVIEWA(15)=IPACK(HCR3,-.14,1)
      IVIEWA(16)=IPACK(HCR5,-.14,0)
      IVIEWA(17)=IPACK(+1.15,-.14,1)
      IVIEWA(18)=IPACK(0.,0.,0)
      GO TO 55
50    IVIEWA(10)=IPACK(X1XA,Y1YA,0)
      IVIEWA(11)=IPACK(X1XA,Y1YR,1)
      IVIEWA(12)=IPACK(.5,REARC2,1)
      IVIEWA(13)=IPACK(X1XFR,Y1YE,0)
      IVIEWA(14)=IPACK(X1XCR,Y1YEE,1)
      IVIEWA(15)=IPACK(-1.15,-.14,0)
      IVIEWA(16)=IPACK(-.5,-.14,1)
      IVIEWA(17)=IPACK(HCR5,-.14,0)
      IVIEWA(18)=IPACK(1.15,-.14,1)
      GO TO 55
51    IVIEWA(10)=IPACK(X1XB,Y1YB,0)
      IVIEWA(11)=IPACK(X1XB,Y1YL,1)
      IVIEWA(12)=IPACK(-.5,REARB2,1)
      IVIEWA(13)=IPACK(X1XEL,Y1YE,0)
      IVIEWA(14)=IPACK(X1XCL,Y1YEE,1)
      IVIEWA(15)=IPACK(-1.15,-.14,0)
      IVIEWA(16)=IPACK(HCR3,-.14,1)
      IVIEWA(17)=IPACK(.5,-.14,0)
      IVIEWA(18)=IPACK(1.15,-.14,1)
55    CONTINUE
      IVIEWA(19)=IPACK(YAWFX,YAWPY,0)
      IVIEWA(20)=IPACK(YAWPX,YAWPY,1)
```

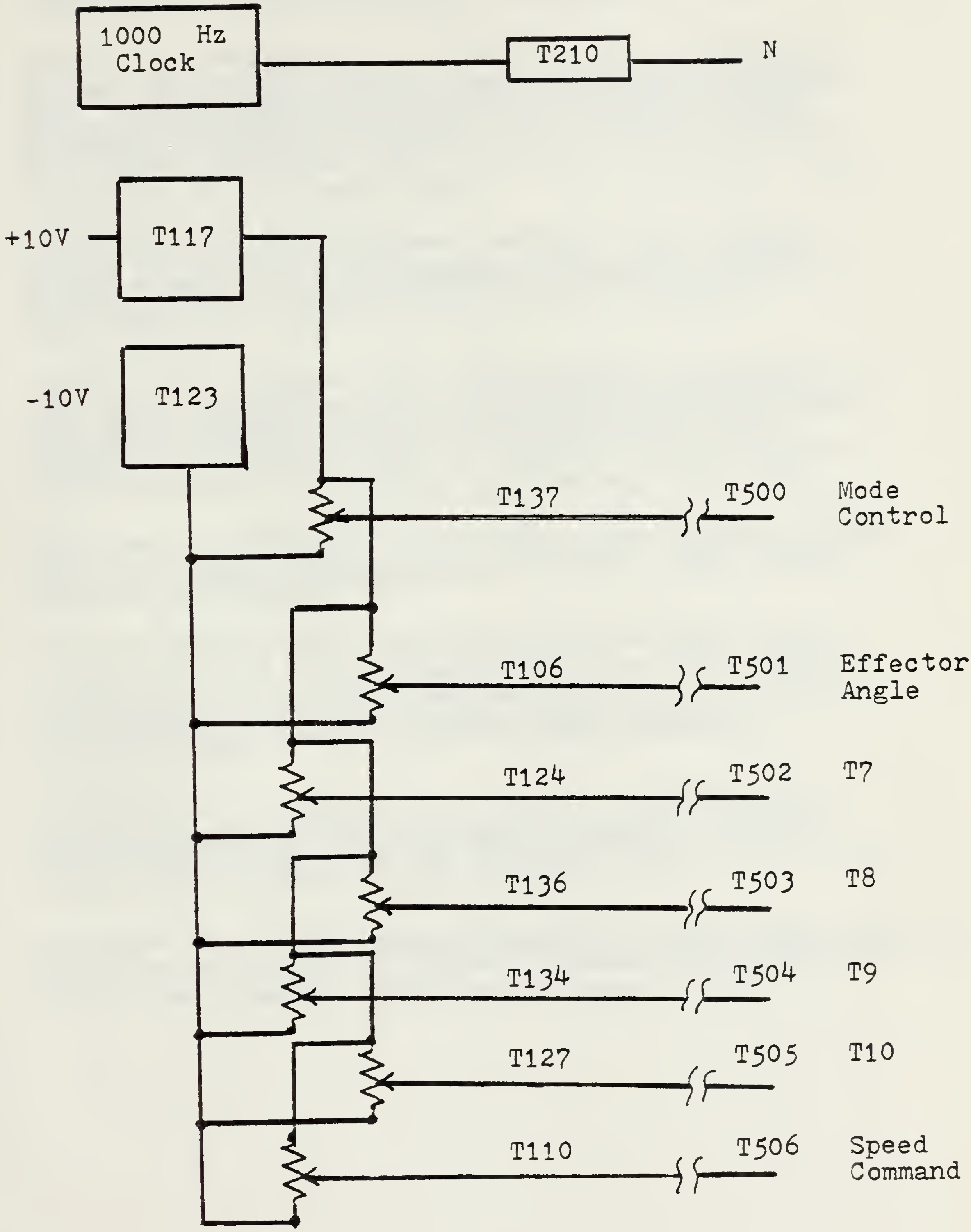
```
      IVIEWA(21)=IPACK(YAWPX,YAWPY,1)
      IVIEWA(22)=IPACK(YAWPX,YAWPY,1)
      IVIEWA(23)=IPACK(ROAD1X,ROAD1Y,0)
      IVIEWA(24)=IPACK(ROAD4X,ROAD4Y,1)
      IVIEWA(25)=IPACK(ROAD2X,ROAD2Y,1)
      IVIEWA(26)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(27)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(28)=IPACK(ROAD1X,ROAD1Y,1)
      IVIEWA(29)=IPACK(ROAD4X,ROAD4Y,0)
      IVIEWA(30)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(31)=0
      CALL GRAPHO(1,IVIEWA,31,3,IER)
 3333 CONTINUE
      RETURN
      END
      SUBROUTINE CNEMOT
      DIMENSION IARR(7)
      COMMON A41,U,DELT,UDOT,V,VDOT,R,RDOT,P,PDOT,Q,QDOT,THETA,T,AM,
     CCDP,LZ,A11,A21,A31,X,AXST,AYST,
     CZI,CDX,RZ,CO,CDY,WW,A22,RDP,PHI,RDS,FSP,FSS,CDZP,A33,RX,F1,F2,
     CF3,LP,LC,FB,WE,RY,L3,A34,Z,BETA,HYSESX,XST,HYSESY,YST,UMAX,XGX,
     CXDX,LNLL,HEAD,YHIGH,SPDLIN,AICOA,YO,ICOA,SEAST
      COMMON /DATA/IARR
      CDX1=230000.
      CDX2=5575.31
      CDX3=13.46
      CDX4=135000000.
      CDX5=13372.96
      ZI=Z-BETA
      A41=10.0*IARR(2)/2**23
      U=U+DELT*UDOT
      IF(U.GT.30.3)GO TO 4
      FSD=CDX1*((U/30.3)**3)
      GO TO 3
    4 CONTINUE
      IF(U.GT.37.03)GO TO 1
      FSD=635200.59-CDX5*U
      GO TO 3
    1 CONTINUE
      IF(U.GT.63.97)GO TO 2
      FSD=CDX2*U-37721.
      GO TO 3
    2 FSD=CDX3*U**2+CDX4/(1+U**1.5)
    3 CONTINUE
      V=V+DELT*VDOT
      R=R+DELT*RDOT
      P=P+DELT*PDOT
      Q=Q+DELT*QDOT
```

```
      PHI=PHI+DELT*P
      THETA=THETA+DELT*Q
      UDCT=((T/AM)*COS(ZI))-FSE/AM+V*R
      VDOT=((T/AM)*SIN(ZI))-((CDY/AM)*V*ABS(V))-U*R
      RDCT=((-1.0)*(T/RZ)*CC*SIN(ZI))+((CDY/RZ)*V*WW*ABS(V))
     C+(A22*L*V*WW)/RZ
      RDP=6.-5C.*((SIN(PHI))/(COS(PHI)))
      RDS=6.+5C.*((SIN(PHI))/(COS(PHI)))
      FSP=128CCC.*RDP
      FSS=128000.*RDS
      PDCT=(((FSF-FSS)*50.)-(T*(SIN(ZI)))+(CDY*V*(ABS(V))*30.)-
     C(CDZP*V*(ABS(V))*50.)-A33*U*P)/RX
      F1=192000.+3200000.*(SIN(THETA))/(COS(THETA))
      F2=192000.-3200000.*(SIN(THETA))/(COS(THETA))
      F3=CDP*PB*WE*(LD-L3*((SIN(THETA))/(COS(THETA))))
      QDCT=((T*(COS(ZI))*LP)+F3*L3+F2*L3-FSD*LZ-F1*L3-A34*U*Q)/RY
      HYSESX=XST
      HYSESY=YST
      RETURN
      END
```

```
$AC     PZE
        BRM     9SETUP1
        PZE     1
BUF     PZE
        LDA     BUF
        ETR     =C77777
        STA     COMM
        LDA     =7
        LLSA    15
        ADD     =COMM
        STA     CONTR
        LDA     ENDBRM
        STA     040
        EOM     034001
        POT     CONTR
        BRR     AC
ENDBRM  BRM     ENDAD
ENDAD   PZE
        DIR
        EOM     034001
        POT     CONTR
        EIR
        BRC     *ENDAD
COMM    RES     7
CONTR   PZE
        END
```

Appendix D

RTS5D Modified Wiring Diagram



LIST OF REFERENCES

1. Nelson, T.F. III, Real Time Simulation and Control 3000 Ton Surface Effect Ship M. S. Thesis, Naval Postgraduate School, Monterey,California, 1979.

2. Kaplan, P., Bentson, J., Sargent, T.P., A Study of Surface Effect Ship (SES) Craft, Loads and Motions, Part I, Equations of Motion of SES Craft with Six Degrees of Freedom, Contract No. C-1-35055, Report No. 71-84A, Oceanics Inc., Technical Industrial Park, Plainview, N.Y., August 1971.

3. Lockheed Missiles and Space Company, Inc., Surface Effect Ships Aero/Hydrodynamics Technology Design Manual, Vol. I, Contract No. N00024-73-A-0919, Task Order No. 0003, May 1974.

4. Gerba, A.Jr., Thaler, G.J., Development of a Real Time Hybrid Computer Simulation for the 3K-SES, 500F, Data Based Program, Part I- Familiarization and Planning, Progress Report to SESPO, U.S. Naval Postgraduate School, Monterey, California, July 1979.

5. Department of the Navy, Code 1630: PRV, Surface Effect Ship Data Based Maneuvering Simulation, SES Project PMS-304, September,1979.

6. David W. Taylor Naval Ship Research and Development Center, Viars,P.R. Unclassified Letter 1630: PRV:wjj to Gerba,A. U.S. Naval Postgraduate School, Subject: Conditions for Constant Speed 3KSES Maneuvers, 5 December 1979.

7. Greensite, A., Pascol, L., Wiemer,M., 3K SES Ship Control System Design Information Report, Volume 1, CDRL No.E05P, Document No. TER023A, Rohr Marine, Inc., Chula Vista California, 24 July 1978.

8. Sturgeon, M.E., The Array Processor: A Real Time Simulator, Paper Presented at the Summer Computer Simulation Conference, Seattle, Washington, August 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Alex Gerba, Jr., Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Professor George J. Thaler, Code 62Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Mr. H. Robertson PMS 304, Code 141 Surface Effects Ships Project Office P. O. Box 34401 Bethesda, Maryland 20034	6
7.	Commanding Officer Naval Electronic Systems Engineering Center, Vallejo Attn: LT Lee L. Oliphant, USN Vallejo, California 94590	1

Thesis 1)1284
0433 Oliphant
c,1 Real time simulation and control: 3000 ton surface effect ship with negative drag characteristics in sea state.